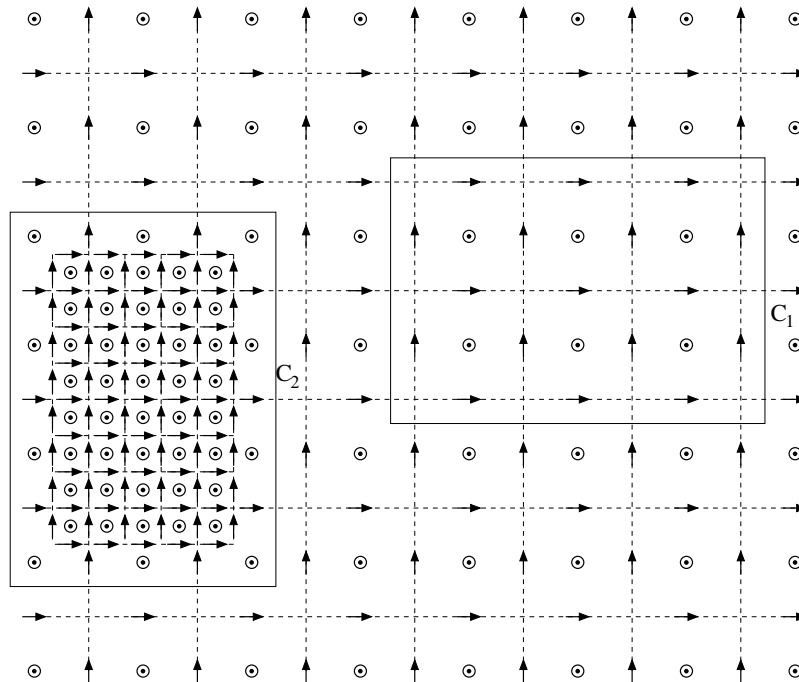


De subdomein FDTD methode

The subdomain FDTD method

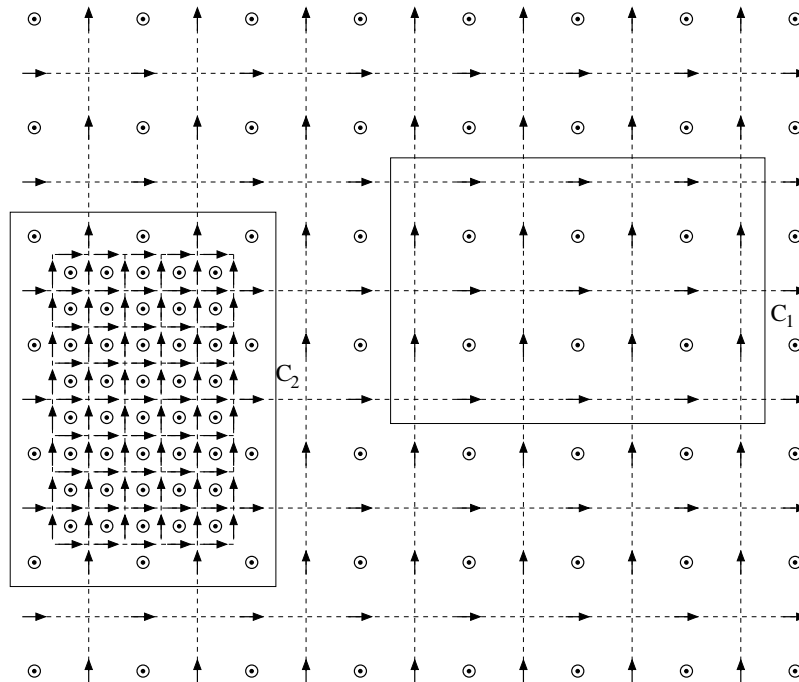
Bart Denecker



De subdomein FDTD methode

The subdomain FDTD method

Bart Denecker



PROMOTOR: PROF. DR. IR. FRANK OLYSLAGER
PROF. DR. IR. LUC KNOCKAERT

UNIVERSITEIT GENT
FACULTEIT TOEGEPASTE WETENSCHAPPEN

VAKGROEP INFORMATIETECHNOLOGIE

SINT-PIETERSNIEUWSTRAAT 41
B-9000 GENT, BELGIË

THESISINFO

Dankwoord

Alhoewel een doctoraat één auteur heeft, wordt het werk door velen gedragen en opgebouwd. Het is het resultaat van een lange tocht vol hindernissen die door veel mensen, zowel bewust als onbewust, gesteund en voortgedreven werd.

Ik zou in eerste instantie mijn promotoren willen bedanken: Frank Olyslager, voor het aanbrengen van de fundamenteën van dit werk, en Luc Knockaert, voor de inbreng van originele ideeën enerzijds en de belangrijkste poot waarop dit werk gesteund is anderzijds. Verder is de steun van Daniël De Zutter, die ingestaan heeft voor al de rest en nog zoveel meer, van groot belang geweest. Van harte bedankt.

Het onderzoek zou nooit mogelijk geweest zijn zonder de ondersteuning die binnen de vakgroep gegeven wordt hierbij gaat mijn hartelijke dank uit naar iedereen, en in het bijzonder naar voorzitter Paul Lagasse.

Verder heeft Steve Van den Berghe in grote mate geholpen door zijn simulator 'SimulateWorld' zodanig te schrijven dat het gemakkelijk kon uitgebreid worden. De vele vragen die hij zonder aarzelen altijd bereid was te beantwoorden hebben me op het goede pad gezet van in het begin. De hulp van Hendrik Rogier en Eric Laermans, altijd paraat om lastige technische problemen op te lossen was een wezenlijke ondersteuning. Verder bedank ik voor het herlezen Gunther Lippens, Davy Pissoort, en voor de nachtelijke steun Pia Perriello, voor de gedrevenheid Stephan Van Damme en voor de relativering Dries Vande Ginste. Collega's en ex-collega's hebben ook bijgedragen tot aangename sfeer binnen de vakgroep doorheen de jaren, hierbij denk ik aan Ann Franchois, Ward Wallyn, Henk Derudder, Bernard De Backer, Jan De Geest, Liestbeth Vandamme en Jegannathan Srinivasan.

De hartelijke ontvangst in Kiel door Ludger Klinkenbusch en de mensen van zijn groep, Sigrid, Jens, Claus-Christian, Micha, Jülf en Simona zal mij lang bijblijven.

Ik dank ook José Antonio Pereda en Luis Antonio Vielva, die mij, meer dan vier jaar geleden, in dit onderzoeksdomein binnengeleid hebben en mij

in Santander de beginselen ervan hebben bijgebracht. Deze laatste weken ben ik terug in hun vaarwater terecht gekomen, de kring is rond.

Tot slot wil ik nog al mijn vrienden en familie bedanken, ik had eerst gedacht om ze één voor één te overlopen en speciaal te bedanken. Maar om te vermijden dat deze bladzijden meer plaats innemen dan de wetenschappelijke inhoud heb ik het gehouden bij een aparte ongeveer 500 bladzijden tellende bijlage die op aanvraag kan bekomen worden. Ik wens hen oprecht te danken omdat ze er waren en omdat ze vooral niet onderlegd zijn in hetgeen ik vier jaar aan gewerkt heb.

Oktober 2002
Bart Denecker

Inhoudsopgave

| | |
|-------------------------------------------------------------|-----------|
| Nederlandstalige Samenvatting | xv |
| 1 Inleiding | xv |
| 1.1 Achtergrond | xv |
| 1.2 Motivering | xvi |
| 2 Ruimtelijke discretisatie | xviii |
| 2.1 Het uniform orthogonaal rooster | xviii |
| 2.2 Verfijning | xx |
| 2.3 Toestandsmodel | xxi |
| 2.4 Subdomein | xxii |
| 3 Tijdsdiscretisatie | xxv |
| 3.1 Standaard FDTD | xxv |
| 3.2 Subdomein FDTD | xxvii |
| 4 Gereduceerde orde modellering (ROM) | xxxiii |
| 4.1 Algemene achtergrond | xxxiii |
| 4.2 Het Laguerre-SVD reductiealgoritme | xxxv |
| 4.3 Algoritmen | xxxvii |
| 5 Stabiliteit | xxxix |
| 5.1 Spatiale reciprociteit | xl |
| 5.2 Subdomein vergelijkingen | xli |
| 6 Numerieke voorbeelden | xlii |
| 6.1 Veralgemeende subdomein FDTD methode | xliii |
| 6.2 De subdomein FDTD methode | xlvi |
| 7 Besluit | xlix |
| Bibliografie | 1 |
| 1 Introduction | 1 |
| 1 Maxwell's equations | 1 |
| 2 Numerical techniques | 2 |
| 3 The Finite-Difference Time-Domain (FDTD) method | 4 |
| 4 Objectives | 7 |

| | | |
|----------|---------------------------------------------------------|-----------|
| 5 | Outline of this work | 8 |
| | Bibliography | 11 |
| 2 | Finite differences: the spatial problem | 15 |
| 1 | Introduction | 15 |
| 2 | Uniform orthogonal grid | 16 |
| 2.1 | Introduction | 16 |
| 2.2 | Generalized 2D Maxwell's equations | 16 |
| 2.3 | The grid | 20 |
| 2.4 | Discretizing space | 22 |
| 2.5 | Properties of spatially discretized equations | 26 |
| 3 | Subgridding | 32 |
| 3.1 | Introduction | 32 |
| 3.2 | Subgridding: first attempt | 33 |
| 3.3 | Odd refinement ratio | 35 |
| 3.4 | Final grid | 37 |
| 3.5 | Accuracy | 38 |
| 3.6 | Dual grid | 41 |
| 4 | State-space models | 42 |
| 4.1 | Introduction | 42 |
| 4.2 | General aspects | 43 |
| 4.3 | Finite difference grids | 44 |
| | Bibliography | 60 |
| 3 | Finite differences: the temporal problem | 63 |
| 1 | Introduction | 63 |
| 2 | Standard FDTD method | 64 |
| 2.1 | Introduction | 64 |
| 2.2 | Scalar notation | 65 |
| 2.3 | Matrix notation | 69 |
| 2.4 | Computational considerations | 72 |
| 3 | Subdomain FDTD method | 72 |
| 3.1 | Subdomain types | 72 |
| 3.2 | Subdomain time discretization | 73 |
| 3.3 | Computational considerations | 77 |
| 3.4 | The subdomain FDTD method generalized | 78 |
| 4 | Alternating-direction implicit FDTD method | 81 |
| 4.1 | Algorithm | 81 |
| 4.2 | Computational considerations | 83 |
| | Bibliography | 84 |

| | | |
|----------|----------------------------------------------------------------------------------|------------|
| 4 | ROM: Reduced order modeling | 87 |
| 1 | Introduction | 87 |
| 2 | Reduced order modeling: general aspects | 89 |
| 3 | Reduced order modeling algorithms | 93 |
| 3.1 | PVL: Padé via Lanczos | 93 |
| 3.2 | PRIMA: Passive Reduced-Order Interconnect Macromod- eling Algorithm | 98 |
| 3.3 | Laguerre-SVD reduced order modeling | 100 |
| 4 | FDTD and ROM | 102 |
| 4.1 | Literature survey | 102 |
| 4.2 | The subdomain FDTD method | 105 |
| | Bibliography | 110 |
| 5 | Stability | 113 |
| 1 | Introduction | 113 |
| 2 | Spatial discretization related stability | 115 |
| 2.1 | Spatially reciprocal grids | 115 |
| 2.2 | Spatially non-reciprocal grids | 121 |
| 3 | ROM related stability | 133 |
| 3.1 | The reduced order model | 133 |
| 3.2 | Spatially reciprocal grids | 136 |
| 3.3 | Spatially nonreciprocal grids | 137 |
| 4 | Time discretization related stability | 138 |
| 4.1 | Standard FDTD method | 138 |
| 4.2 | ADI-FDTD method | 148 |
| 4.3 | Subdomain FDTD method | 150 |
| 5 | Overview | 156 |
| | Bibliography | 157 |
| 6 | Numerical examples | 161 |
| 1 | Introduction | 161 |
| 2 | Generalized Subdomain FDTD method | 161 |
| 3 | Subdomain FDTD method | 170 |
| 3.1 | Introduction | 170 |
| 3.2 | Lossless free space. | 173 |
| 3.3 | Perfectly conducting thin wire. | 174 |
| 3.4 | Dielectric thin wire. | 177 |
| 3.5 | L-shaped lossy dielectric object. | 179 |
| 3.6 | Curved corner region. | 183 |
| 3.7 | Influence of the interpolations. | 184 |
| 3.8 | Increasing time step | 186 |
| 3.9 | Photonic crystal waveguide | 190 |

| | |
|----------------------------------------|------------|
| Bibliography | 194 |
| 7 Conclusions — Future Research | 197 |
| 1 Conclusions | 197 |
| 2 Future research | 200 |
| Bibliography | 201 |

Nederlandstalige Samenvatting

Nederlandstalige Samenvatting

1 INLEIDING

1.1 Achtergrond

De impact van elektromagnetisme, reeds belangrijk in het verleden, neemt heden ten dage nog verder toe. De redenen daarvoor zijn velerlei: aan de ene kant van het spectrum zijn er problemen met chipinterconnecties. Deze interconnecties gedragen zich hoe langer hoe meer als golfgeleiders, als gevolg van de hogere frequenties van de signalen die eroverheen propageren. Het golfkarakter van deze doorverbindingen komt tot uiting in parasitaire effecten zoals bijvoorbeeld overspraak, waardoor een pseudo-signaal begint te propageren op nabijgelegen interconnecties. Daardoor komt het dat de stroom- en spanningsvergelijkingen van Kirchhoff in de meeste hedendaagse circuits niet meer kloppen en dat dergelijke circuits moeten geanalyseerd worden aan de hand van meer algemene technieken die gebaseerd zijn op de vergelijkingen van Maxwell.

Aan de andere kant van het spectrum, in de optica, gaat door het verder verkleinen van optische componenten de benadering dat de golflengte veel kleiner is dan de karakteristieke afmetingen van de componenten niet langer op. Daardoor dient men zich daar ook hoe langer hoe meer te richten tot een rigoureuze oplossing van de vergelijkingen van Maxwell.

Aangezien de vergelijkingen van Maxwell slechts in een klein aantal gevallen analytisch kunnen opgelost worden steunt men op numerieke technieken om complexe hedendaagse elektromagnetische problemen op te lossen. Verschillende technieken zijn in de loop van de jaren voorgesteld, waarvan de meest bekende en tevens meest populaire zijn: de eindige elementenmethode (*E. finite elements method, FEM*) [1], de randintegraalvergelijkingstechniek (*E. boundary integral equations, BIE*) opgelost met de mo-

mentenmethode (*E. method of moments, MoM*) [2] en de eindige differenties in het tijdsdomein methode (*E. finite difference in time domain, FDTD*) [3]. Elk van deze methodes heeft zijn eigen verdiensten en bestaansredenen. De ene methode slaagt er bijvoorbeeld beter in om problemen met open gebieden te behandelen (MoM), de andere methode kan gemakkelijk problemen met meerdere diëlektrische materialen met complexe geometriën analyseren (FEM), terwijl een derde erin slaagt om niet-lineaire materialen op een eenvoudige manier te behandelen (FDTD).

In dit werk zal enkel de FDTD methode behandeld worden. Heden ten dage is de FDTD techniek een zeer bedrijvig onderzoeksgebied. Dit kan aangetoond worden aan de hand van de statistieken van de webstek, www.fDTD.org, die volledig gewijd is aan deze numerieke techniek en voornamelijk referenties naar artikels in tijdschriften en op conferenties bevat. De databank bevat op dit ogenblik ongeveer 2500 referenties naar artikels in internationale tijdschriften en 162 referenties van doctoraatsthesisen. Het overgrote deel van deze artikels werd de voorbije tien jaar gepubliceerd.

1.2 Motivering

Een belangrijke karakteristiek van de standaard FDTD methode is het gebruik van een uniform rooster: de discretisatie is constant in iedere richting. Dankzij dit uniform rooster slaagt de techniek er in om op een efficiënte manier met een heel groot aantal variabelen, alleszins in vergelijking met FEM en MoM, te werken. Het uniform rooster betekent ook dat de simulatieruimte opgebouwd is uit cellen van constante omvang. Voor een groot aantal applicaties is dit geen probleem. Bio-elektromagnetische toepassingen, waar bijvoorbeeld wordt getracht na te gaan op welke manier de elektromagnetische golven in de hersenen worden geabsorbeerd bij het telefoneren met een GSM, maken gebruik van een mathematisch model van een menselijk lichaam. Dat dit model opgebouwd is uit uniforme cellen, waarbij elke cel zijn eigen elektrische eigenschappen heeft, is een nadeel dat niet opweegt tegen het groot aantal variabelen dat in simulaties kan gebruikt worden. De onzekerheid van het numerieke mensmodel kan toch niet verbeterd worden met andere methodes.

Bij een groot aantal andere problemen zorgt dit uniform rooster wel voor problemen. Vaak is het zo dat de velden sterk variëren in kleine gebieden van het simulatiedomein. Als men deze gebieden correct wenst te modelleren en, als men vertrekt van een uniform rooster, dan dient men de roosterdimensies, ook dikwijls ruimtestap geheten, te verkleinen overeenkomstig de sterk variërende velden. Dit resulteert, in de standaard FDTD

techniek, in een grote toename van de simulatiekost. Aangezien de tijdstap die gebruikt wordt in de FDTD methode evenredig is met de ruimtestap heeft een halvering van de ruimtestap met een factor twee een toename van 16 in reken capaciteit tot gevolg voor 3D problemen. Dit is in veel gevallen ontoelaatbaar.

Een aantal technieken om aan deze tekortkomingen, verbonden aan het uniform rooster, tegemoet te komen zijn tot nu toe voorgesteld. Twee klassen zijn vooral populair: het lokaal verfijnen van het rooster en het gebruik van subcellen. Een eerste techniek is lokaal het rooster verfijnen. Deze aanpak tracht een fijn rooster en een grof rooster met elkaar te combineren. De tijdstappen die gebruikt worden in beide roosters zijn meestal niet aan elkaar gelijk. Dit betekent dat het een hele klus is om beide roosters aan elkaar te koppelen in het tijdsdomein. Dikwijls heeft dit tot gevolg dat men gebruik moet maken van extrapolaties [4] of van een artificiële golfvergelijking [5], met een ingewikkeld algoritme en een zekere mate van onnauwkeurigheid tot gevolg. De aanpak heeft echter als voordeel dat hij bijzonder algemeen is. Een tweede techniek om de beperkingen van een uniform rooster op te vangen zijn subcel technieken. In deze aanpak tracht men, steunend op het gekend statisch gedrag rond specifieke kleine geometrieën, de FDTD iteratievergelijkingen in de buurt van het element aan te passen en zo deze gedragingen mee te simuleren. Deze aanpak werkt echter enkel in een aantal specifieke gevallen en is dus geenszins algemeen. Het bekendst is het subcel model om een dunne perfect geleidende draad te simuleren [6]. Het voordeel is de eenvoud van het algoritme, eenmaal men de nieuwe vergelijkingen opgesteld heeft.

In dit werk wordt de subdomein FDTD techniek voorgesteld. Deze techniek bevat de beste eigenschappen van beide aanpakken: enerzijds is de techniek zeer algemeen en anderzijds is het simulatiealgoritme zeer eenvoudig. Als een uitbreiding van deze aanpak wordt ook een veralgemening voorgesteld: de veralgemeende subdomein FDTD techniek. Beide technieken worden onderbouwd door duidelijk voor ogen te houden dat bij de FDTD techniek de spatiale of ruimtelijke discretisatie en de tijdsdiscretisatie twee aparte aspecten in het afleiden van de vergelijkingen zijn. Daarnaast, om de aanpak efficiënt te maken, wordt er gesteund op een nieuwe mathematische techniek die toelaat om toestandsmodellen te benaderen door modellen van gereduceerde orde (*E. reduced order model, ROM*).

2 RUIMTELIJKE DISCRETISATIE

2.1 Het uniform orthogonaal rooster

De standaard eindige differenties in het tijdsdomein methode (*E. Finite Difference Time Domain, FDTD*) is gebaseerd op een puntsgewijze discretisatie van de vergelijkingen van Maxwell. De discretisaties vervangen afgeleiden door centrale differentie benaderingen:

$$\left. \frac{df(x)}{dx} \right|_{x=x_0} = \frac{f(x_0 + \Delta/2) - f(x_0 - \Delta/2)}{\Delta} + \mathcal{O}(\Delta^2) \quad (1a)$$

en indien nodig functiewaarden door functiewaarden in naburige punten:

$$f(x_0) = \frac{f(x_0 + \Delta/2) + f(x_0 - \Delta/2)}{2} + \mathcal{O}(\Delta^2) \quad (1b)$$

Beide benaderingen zijn tweede orde nauwkeurig: $\mathcal{O}(\Delta^2)$.

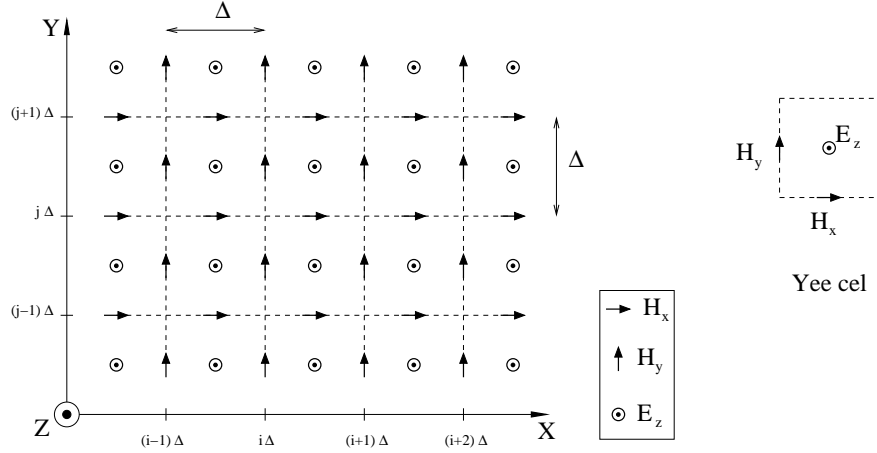
Beschouwen we nu de vergelijkingen van Maxwell voor isotrope, tijdsinvariante en lineaire materialen, waarbij verondersteld wordt dat noch de excitatie noch de bestudeerde geometrie varieert in de z -richting, dan blijkt dat de vergelijkingen uiteenvallen in twee onafhankelijke deelp Problemen. Enerzijds het TM-probleem (*E. transverse magnetic*), waarbij enkel de veldcomponenten H_x , H_y en E_z aanwezig zijn en anderzijds het TE-probleem (*E. transverse electric*), waarbij enkel de veldcomponenten E_x , E_y en H_z aanwezig zijn. Beide gevallen zijn mathematisch aan elkaar gelijk en we zullen ons dan ook beperken tot het bestuderen van één enkel geval: het TM-probleem. De vergelijkingen in dit geval zijn:

$$\mu(\mathbf{r}) \frac{\partial H_x(\mathbf{r}, t)}{\partial t} = -\frac{\partial E_z(\mathbf{r}, t)}{\partial y} - \sigma_m(\mathbf{r}) H_x(\mathbf{r}, t) \quad (2a)$$

$$\mu(\mathbf{r}) \frac{\partial H_y(\mathbf{r}, t)}{\partial t} = \frac{\partial E_z(\mathbf{r}, t)}{\partial x} - \sigma_m(\mathbf{r}) H_y(\mathbf{r}, t) \quad (2b)$$

$$\epsilon(\mathbf{r}) \frac{\partial E_z(\mathbf{r}, t)}{\partial t} = -\frac{\partial H_x(\mathbf{r}, t)}{\partial y} + \frac{\partial H_y(\mathbf{r}, t)}{\partial x} - \sigma_e(\mathbf{r}) E_z(\mathbf{r}, t) \quad (2c)$$

In de standaard FDTD methode wordt de discretisatievergelijking (1a) toegepast op de spatiale aspecten van deze 2D vergelijkingen. Dit wordt gedaan door de probleemruimte te beschouwen als zijnde opgebouwd uit cellen van constante omvang, vandaar uniform rooster, die daarenboven rechthoekig zijn, vandaar een orthogonaal rooster, en door binnen deze cellen de veldcomponenten te schranken. Deze cellen worden Yee cellen geheten, naar K. S. Yee die dit geschrinkt rooster in 1966 voor het eerst voorstelde [7]. Wij zullen ons beperken tot vierkante cellen, $\Delta_x = \Delta_y = \Delta$.



Figuur 1: Een stuk van een uniform orthogonaal rooster, met rechts de basis bouwsteen: de Yee cel. De geschrante positie van de verschillende veldcomponenten is duidelijk zichtbaar.

In Fig. 1, is een stuk van een dergelijk rooster te zien. Iedere veldvariabele is van één van de volgende vormen: $H_x(t)|_{(i+1/2)\Delta, j\Delta}$, $H_y(t)|_{i\Delta, (j+1/2)\Delta}$ en $E_z(t)|_{(i+1/2)\Delta, (j+1/2)\Delta}$, waarbij $i, j \in \mathbb{Z}$. Om de notatie te verlichten wordt ten eerste de tijdsafhankelijkheid weggelaten en wordt ten tweede de Δ -afhankelijkheid weggelaten: $H_x(t)|_{(i+1/2)\Delta, j\Delta} = H_x|_{i+1/2, j}$. Door de gepaste vergelijking uit (2) op de locatie van de betreffende veldcomponent te discretiseren krijgen we dan:

$$\mu|_{i+1/2, j} \frac{dH_x|_{i+1/2, j}}{dt} = -\frac{1}{\Delta} \begin{bmatrix} E_z|_{i+1/2, j+1/2} \\ -E_z|_{i+1/2, j-1/2} \end{bmatrix} - \sigma_m|_{i+1/2, j} H_x|_{i+1/2, j} \quad (3a)$$

$$\mu|_{i, j+1/2} \frac{dH_y|_{i, j+1/2}}{dt} = \frac{1}{\Delta} \begin{bmatrix} E_z|_{i+1/2, j+1/2} \\ -E_z|_{i-1/2, j+1/2} \end{bmatrix} - \sigma_m|_{i, j+1/2} H_y|_{i, j+1/2} \quad (3b)$$

$$\epsilon|_{i+1/2, j+1/2} \frac{dE_z|_{i+1/2, j+1/2}}{dt} = \frac{1}{\Delta} \begin{bmatrix} -H_x|_{i+1/2, j+1} + H_x|_{i+1/2, j} \\ +H_y|_{i+1, j+1/2} - H_y|_{i, j+1/2} \end{bmatrix} - \sigma_e|_{i+1/2, j+1/2} E_z|_{i+1/2, j+1/2} \quad (3c)$$

Het belang van het schranken van de veldcomponenten wordt op deze manier duidelijk: de locatie van de veldvariabelen aan de linkerkant corresponderen met die aan de rechterkant in de andere vergelijkingen. Het valt ook op te merken dat iedere vergelijking enkel de veldvariabele zelf en buurtveldvariabelen bevat.

Een belangrijke eigenschap van deze vergelijkingen is spatiale reciprociteit. Daartoe definiëren we de vergelijking van een veldvariabele als die ver-

gelijking waarbij de veldvariabele links staat in (3), of met andere woorden als die vergelijking waarin de tijdsafgeleide van de veldvariabele voorkomt. Dan betekent spatiale reciprociteit dat als in de vergelijking van een veldvariabele een andere veldvariabele gebruikt wordt dat dan omgekeerd in de vergelijking van die andere veldvariabele de eerste veldvariabele gebruikt wordt:

$$\mu|_{i+1/2,j} \frac{dH_x|_{i+1/2,j}}{dt} = -\frac{1}{\Delta} E_z|_{i+1/2,j+1/2} \dots \quad (4)$$

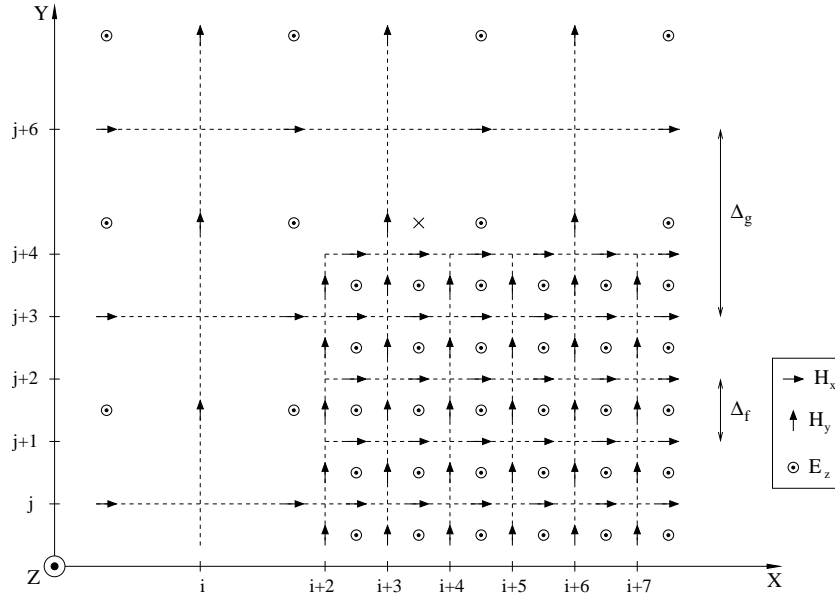
$$\epsilon|_{i+1/2,j+1/2} \frac{dE_z|_{i+1/2,j+1/2}}{dt} = \frac{1}{\Delta} H_x|_{i+1/2,j} \quad (5)$$

De coëfficiënten in beide vergelijkingen zijn gelijk maar tegengesteld. Deze eigenschap is belangrijk om garanties betreffende stabiliteit van het uiteindelijk algoritme te kunnen geven [8, 9].

2.2 Verfijning

Zoals in de inleiding gezegd zijn we niet enkel geïnteresseerd in uniforme roosters. Door in bepaalde gebieden een fijner rooster te gebruiken en voor de rest het uniforme, grovere (*E. coarse*) rooster te gebruiken bekomen we heel wat meer vrijheden. Deze techniek heet verfijning (*E. subgridding*). Hier stellen we een verfijningstechniek voor die tweede orde nauwkeurig is. Daartoe worden een aantal beperkingen ingevoerd. De verfijningsverhouding, $r = \Delta_g / \Delta_f$, dit is de verhouding van de ruimtestap in het grof rooster tot de ruimtestap in het fijn rooster, dient oneven gekozen te worden. De verfijning is dezelfde in beide richtingen en het fijn rooster omvat een rechthoekig gebied waarbij de randen samenvallen met de randen van de grove cellen uitgebreid met een laag van $(r - 1)/2$ fijne cellen. In Fig. 2 is een hoekgedeelte van een dergelijk verfijnd gebied te zien, de posities op de assen zijn evenredig met Δ_f . Het verfijnd gebied omvat een aantal grove cellen, $x \geq i + 3$ en $y \leq j + 3$, met $(r - 1)/2 = 1$ laag fijne cellen daaromheen.

Het is duidelijk dat voor veldvariabelen niet gelegen aan de overgang van het fijn rooster naar het grof rooster de vergelijkingen (3), met gepaste Δ , opgaan. Voor de veldvariabelen gelegen in het grof rooster en aan de overgang is een tweede orde nauwkeurige vergelijking ook gemakkelijk te bekomen aangezien, bijvoorbeeld voor $E_z|_{i+3/2,j+3/2}$, de ontbrekende veldvariabele ook bestaat in het fijn rooster, dit is dan $H_y|_{i+3,j+3/2}$. Omgekeerd gaat dit helaas niet op. Als oplossing dient er dan gebruik gemaakt te worden van interpolatie. In de vergelijking van $H_x|_{i+7/2,j+4}$ dient een geïnterpoleerde waarde van E_z op de locatie van het kruisje (×) gecombi-



Figuur 2: Verfijning met een oneven verfijningsverhouding, $r = 3$. Het fijn rooster en het grof rooster kunnen duidelijk onderscheiden worden.

neerd te worden met $E_z|_{i+7/2, j+7/2}$ om de afgeleide in de y -richting te benaderen. Deze geïnterpoleerde waarde op \times wordt berekend aan de hand van $E_z|_{i+3/2, j+9/2}$, $E_z|_{i+9/2, j+9/2}$ en $E_z|_{i+15/2, j+9/2}$. Het kan aangetoond worden dat deze werkwijze resulteert in een benadering die tweede orde nauwkeurig is. Verder dient er opgemerkt te worden dat deze werkwijze het verlies van spatiale reciprociteit tot gevolg heeft.

2.3 Toestandsmodel

Een lineair toestandsmodel (*E. state-space model*) beschrijft een lineair, tijdsinvariant systeem aan de hand van een, mogelijks groot, aantal eerste orde differentiaal vergelijkingen. Mathematisch wordt een toestandsmodel beschreven als volgt:

$$\begin{cases} \mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{L}^T \mathbf{x} \end{cases} \quad (6)$$

Hierbij is \mathbf{x} een vector van dimensie N die de toestandsveranderlijken bevat, de vector \mathbf{u} bevat de ingangsvariabelen ($\dim[\mathbf{u}] = p$) en de vector \mathbf{y} bevat de uitgangsveranderlijken ($\dim[\mathbf{y}] = m$). De matrices \mathbf{C} en \mathbf{G} hebben

dimensie $N \times N$, de ingangsmatrix \mathbf{B} heeft dimensie $N \times p$ en de uitgangsmatrix \mathbf{L} heeft dimensie $N \times m$. Hierbij bevatten al deze matrices reële getallen en zijn alle veranderlijken reëel. Een toestandsmodel (6) beschrijft hoe een systeem gestuurd door een aantal ingangsveranderlijken, vector \mathbf{u} , en gecontroleerd door een aantal uitgangsveranderlijken, vector \mathbf{y} , zich gedraagt aan de hand van een aantal interne variabelen, vector \mathbf{x} . In ons geval geldt meestal $N \gg p, m$.

Voor een gediscretiseerde probleemruimte, afgesloten door een perfect geleidend materiaal, $E_z = 0$, kan het toestandsmodel in blokvorm geschreven worden als:

$$\begin{bmatrix} \mathbf{D}_\mu & 0 \\ 0 & \mathbf{D}_\epsilon \end{bmatrix} \begin{bmatrix} \dot{\mathbf{h}} \\ \dot{\mathbf{e}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma_m} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{D}_{\sigma_e} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \mathbf{e} \end{bmatrix} \quad (7)$$

De vector \mathbf{e} bevat alle elektrische veldvariabelen en de vector \mathbf{h} bevat alle magnetische veldvariabelen. De matrices \mathbf{D} zijn diagonaal, waarbij \mathbf{D}_μ de μ -waarden en \mathbf{D}_{σ_m} de σ_m -waarden van de corresponderende magnetische veldvariabelen bevat, en \mathbf{D}_ϵ de ϵ -waarden en \mathbf{D}_{σ_e} de σ_e -waarden van de corresponderende elektrische veldvariabelen bevat. De matrices \mathbf{G}_{12} en \mathbf{G}_{21} bevatten de coëfficiënten ten gevolge van de discretisatie van de afgeleiden. Als dit toestandsmodel gelinkt is met een probleemruimte met een uniform, orthogonaal rooster, dan geldt $\mathbf{G}_{12} = -\mathbf{G}_{21}^T$. Dit is een rechtstreeks gevolg van de spatiale reciprociteit. Het is dan duidelijk dat, bij gebruik van de roosterverfijning zoals voorgesteld hiervoor, er geldt dat $\mathbf{G}_{12} \neq -\mathbf{G}_{21}^T$, aangezien de spatiale reciprociteit in dat geval niet verzekerd is.

In (7) worden de ingangen en de uitgangen van het systeem niet beschouwd, in het algemeen is een ingang een of andere excitatie, b.v. de grootte van een invallende vlakke golf, en is een uitgang een fysische grootte die men wenst te kennen, b.v. het verstrooid veld ten gevolge van de ingevallen vlakke golf. De ingangen en uitgangen van het gesimuleerd probleem zijn niet van belang bij onze analyse aangezien ze het karakter van het systeem niet beïnvloeden. Meer details in verband met bronnen en de resulterende velden en hoe deze te interpreteren kan in [10] gevonden worden.

2.4 Subdomein

Het is nu mogelijk om het rooster in subdomeinen onder te verdelen. Met ieder subdomein (*E. subdomain*) kan er dan een toestandsmodel geassocieerd worden. In Fig. 3 is een uniform rooster te zien dat opgedeeld is in twee subdomeinen. De opdeling is gebeurd aan de hand van een snede.

Het toestandsmodel, in blokform, voor beide deelsystemen is:

$$\begin{cases} \begin{bmatrix} \mathbf{D}_{\mu,\alpha} & 0 \\ 0 & \mathbf{D}_{\epsilon,\alpha} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{h}}_\alpha \\ \dot{\mathbf{e}}_\alpha \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma_m,\alpha} & \mathbf{G}_{12,\alpha} \\ -\mathbf{G}_{12,\alpha}^T & \mathbf{D}_{\sigma_e,\alpha} \end{bmatrix} \begin{bmatrix} \mathbf{h}_\alpha \\ \mathbf{e}_\alpha \end{bmatrix} + \mathbf{B}_\alpha \mathbf{u}_\alpha \\ \mathbf{y}_\alpha = \mathbf{L}_\alpha^T \begin{bmatrix} \mathbf{h}_\alpha \\ \mathbf{e}_\alpha \end{bmatrix} \end{cases} \quad (8)$$

met $\alpha = 1$ voor subdomein 1 en $\alpha = 2$ voor subdomein 2. De ingangs- en uitgangsveranderlijken van beide deelsystemen geven de koppeling weer van beide deelsystemen, en zijn, in tegenstelling tot de ingangs- en uitgangsveranderlijken van het ganse systeem, wel van belang voor ons. De ingangsveranderlijken van subdomein 1 zijn gelijk aan de uitgangsveranderlijken van subdomein 2, dit zijn de veldvariabelen die aangeduid zijn met ∇ in Fig. 3. De ingangsveranderlijken van subdomein 2 zijn de uitgangsveranderlijken van subdomein 1, dit zijn de veldvariabelen die aangeduid zijn met \square in Fig. 3. De uitgangsveranderlijken zijn telkens de veldvariabelen op de rand van het betreffende subdomein. Mathematisch is dit:

$$\mathbf{u}_1 = \mathbf{y}_2 \quad (9a)$$

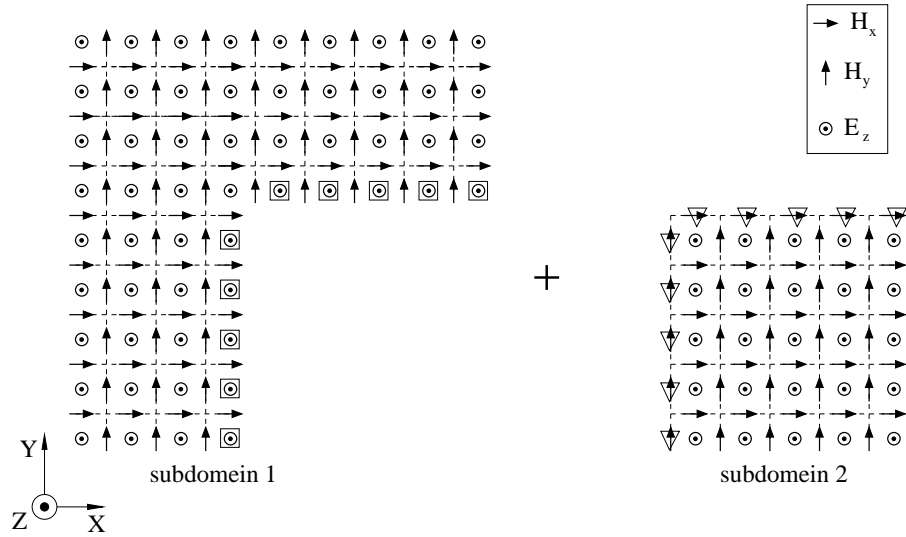
$$\mathbf{u}_2 = \mathbf{y}_1 \quad (9b)$$

Het opsplitsen in subdomeinen in Fig. 3 is zodanig gebeurd dat de veldvariabelen op de rand van het ene subdomein enkel elektrische veldcomponenten zijn, dit subdomein wordt dan een *E*-type subdomein geheten (subdomein 1). De veldvariabelen van \mathbf{y} zijn die variabelen die buien hebben in het andere subdomein. De veldvariabelen op de rand van het andere subdomein zijn magnetische veldcomponenten, dit is dan een *H*-type subdomein (subdomein 2).

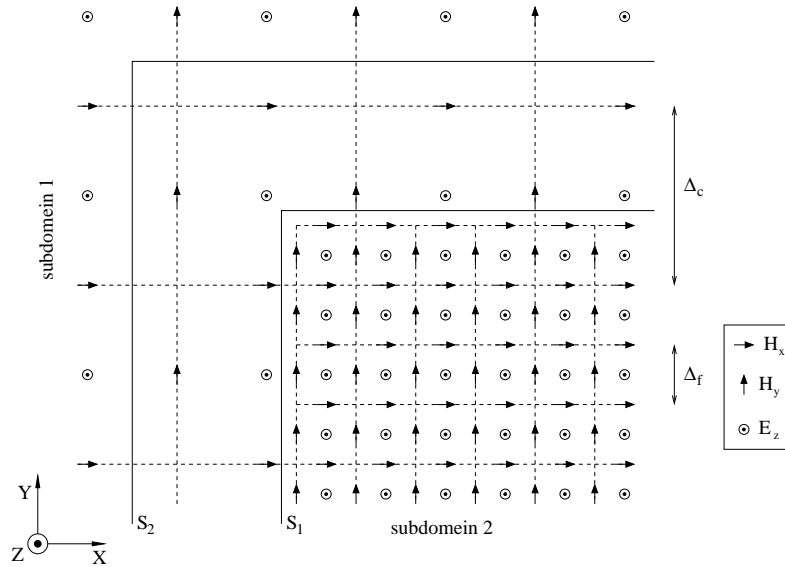
De blokken op de nevendiagonaal van de \mathbf{G} -matrix zijn elkaars tegengestelde, $\mathbf{G}_{21,\alpha} = -\mathbf{G}_{12,\alpha}^T$, want spatiale reciprociteit geldt binnen elk subdomein. Daarenboven geldt er $\mathbf{B}_1 \mathbf{L}_2^T = -(\mathbf{B}_2 \mathbf{L}_1^T)^T$. Dit kan bewezen worden door het toestandsmodel van het volledige systeem terug samen te stellen.

Indien in de probleemruimte gebruik gemaakt wordt van verfijning dan zijn er twee keuzes van opsplitsen die van belang zijn. De eerste keuze plaatst de snede op de grens van het fijne rooster en het grove rooster. In Fig. 4 komt dit overeen met S_1 . Aangezien spatiale reciprociteit niet langer geldig is over de grens tussen het fijn en het grof rooster heen, geldt er voor subdomein 1 en subdomein 2 dat $\mathbf{G}_{12,\alpha} = -\mathbf{G}_{21,\alpha}^T$, maar de relatie tussen ingangs- en uitgangsveranderlijken geldt echter niet meer: $\mathbf{B}_1 \mathbf{L}_2^T \neq -(\mathbf{B}_2 \mathbf{L}_1^T)^T$.

Een tweede keuze is de snede, S_2 , een cel verder te kiezen. Voor de duidelijkheid: wat ook de keuze is de relatie (9) blijft gelden. In dit geval



Figuur 3: Een uniform orthogonaal rooster dat opgesplitst is in twee subdomeinen. De uitgangsvariabelen van subdomein 1 zijn aangeduid met \square en de uitgangsvariabelen van subdomein 2 zijn aangeduid met ∇ .



Figuur 4: Twee mogelijke keuzes om een rooster met verfijning, $r = 3$ op te splitsen. De eerste keuze correspondeert met snede S_1 , de tweede met S_2 .

is, $\mathbf{B}_1 \mathbf{L}_2^T = -(\mathbf{B}_2 \mathbf{L}_1^T)^T$ en de relatie, $\mathbf{G}_{12} = -\mathbf{G}_{21}^T$, geldt voor subdomein 1 maar niet voor subdomein 2.

Beide keuzes zijn zodanig gemaakt dat de grootte van de vectoren \mathbf{u} en \mathbf{y} zo klein mogelijk is. De koppeling tussen beide subdomeinen gebeurt met een minimaal aantal veranderlijken.

3 TIJDSDISCRETISATIE

3.1 Standaard FDTD

De eindige differenties in het tijdsdomein methode of FDTD methode is zoals de naam het zegt, een tijdsdomeinmethode. Concreet betekent dit dat ook de tijdsafgeleiden door een gediscrètiseerde versie benaderd worden. Normaal gezien wordt dit uitgelegd aan de hand van de drie soorten vergelijkingen: voor de H_x , H_y en E_z componenten. Hier zullen we dit uitleggen met behulp van (7), $\mathbf{G}_{21} = -\mathbf{G}_{12}^T$. Eerst wordt het toestandsmodel met de verschillende blokken uitgeschreven:

$$\mathbf{D}_\mu \dot{\mathbf{h}} = -\mathbf{G}_{12} \mathbf{e} - \mathbf{D}_{\sigma_m} \mathbf{h} \quad (10a)$$

$$\mathbf{D}_\epsilon \dot{\mathbf{e}} = \mathbf{G}_{12}^T \mathbf{h} - \mathbf{D}_{\sigma_e} \mathbf{e} \quad (10b)$$

Vervangen we de tijdsafgeleide van iedere veldvariabele in $\dot{\mathbf{e}}$ en in $\dot{\mathbf{h}}$ door een gediscrètiseerde benadering (1a), dan wordt dit in vectorvorm:

$$\dot{\mathbf{h}}|n \simeq \frac{\mathbf{h}|^{n+1/2} - \mathbf{h}|^{n-1/2}}{\Delta_t} \quad (11a)$$

$$\dot{\mathbf{e}}|^{n+1/2} \simeq \frac{\mathbf{e}|^{n+1} - \mathbf{e}|^n}{\Delta_t} \quad (11b)$$

waar Δ_t de tijdstap voorstelt. Belangrijk hierbij is dat het tijdstip waarop de magnetische veldvariabelen en de elektrische veldvariabelen gediscrètiseerd worden een halve tijdstap uit elkaar liggen, $t = n\Delta_t$ en $t = (n + 1/2)\Delta_t$ respectievelijk. Als hetzelfde principe toegepast wordt voor de verliestermen, maar dan met (1b), krijgen we:

$$\mathbf{h}|n \simeq \frac{\mathbf{h}|^{n+1/2} + \mathbf{h}|^{n-1/2}}{2} \quad (12a)$$

$$\mathbf{e}|^{n+1/2} \simeq \frac{\mathbf{e}|^{n+1} + \mathbf{e}|^n}{2} \quad (12b)$$

Wanneer dit allemaal in (10) ingevoerd wordt, dan bekomt men de standaard FDTD vergelijkingen in matrix vorm:

$$\mathbf{h}|^{n+1/2} = \left(\frac{\mathbf{D}_\mu}{\Delta_t} + \frac{\mathbf{D}_{\sigma_m}}{2} \right)^{-1} \left(\frac{\mathbf{D}_\mu}{\Delta_t} - \frac{\mathbf{D}_{\sigma_m}}{2} \right) \mathbf{h}|^{n-1/2} - \left(\frac{\mathbf{D}_\mu}{\Delta_t} + \frac{\mathbf{D}_{\sigma_m}}{2} \right)^{-1} \mathbf{G}_{12} \mathbf{e}^n \quad (13a)$$

$$\mathbf{e}|^{n+1} = \left(\frac{\mathbf{D}_\epsilon}{\Delta_t} + \frac{\mathbf{D}_{\sigma_e}}{2} \right)^{-1} \left(\frac{\mathbf{D}_\epsilon}{\Delta_t} - \frac{\mathbf{D}_{\sigma_e}}{2} \right) \mathbf{e}|^n + \left(\frac{\mathbf{D}_\epsilon}{\Delta_t} + \frac{\mathbf{D}_{\sigma_e}}{2} \right)^{-1} \mathbf{G}_{12}^T \mathbf{h}|^{n+1/2} \quad (13b)$$

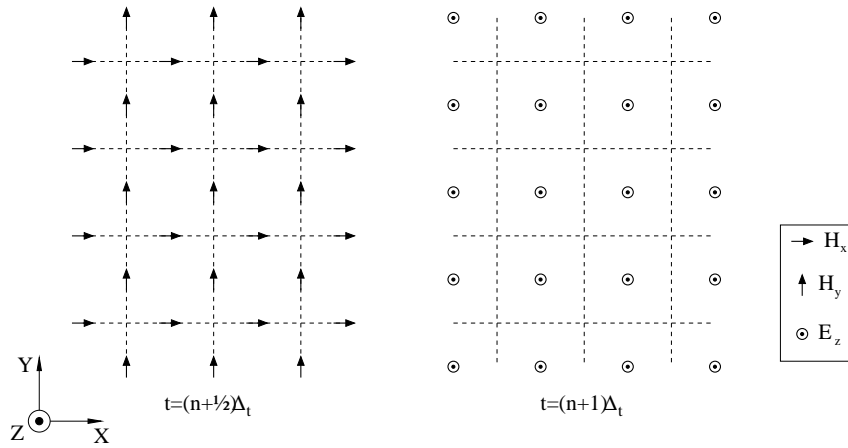
De discretisaties die geleid hebben tot deze vergelijkingen zijn tweede orde nauwkeurig. Aangezien in deze vergelijkingen de \mathbf{D} -matrices diagonaal zijn, geldt er, b.v. voor de eerste vergelijking, dat

$$\left(\frac{\mathbf{D}_\mu}{\Delta_t} + \frac{\mathbf{D}_{\sigma_m}}{2} \right)^{-1} \left(\frac{\mathbf{D}_\mu}{\Delta_t} - \frac{\mathbf{D}_{\sigma_m}}{2} \right) \quad \text{en} \quad \left(\frac{\mathbf{D}_\mu}{\Delta_t} + \frac{\mathbf{D}_{\sigma_m}}{2} \right)^{-1} \quad (14)$$

ook diagonaal zijn, en dat de vergelijkingen in (13) zeer weinig bewerkingen vragen. De nieuwe waarde van een veldvariabele kan berekend worden door één vermenigvuldiging met zijn oude waarde en dit op te tellen met een algebraïsche som van de waarden van naburige veldvariabelen, bepaald door matrix \mathbf{G}_{12} . Het iteratieschema van de standaard FDTD methode gaat dan als volgt:

1. start met $n = 0$ en zet alle veldwaarden op nul
2. gebruik (13a) om de nieuwe waarden van de magnetische veldvariabelen te berekenen en voeg eventuele bronwaarden toe
3. gebruik (13b) om de nieuwe waarden van de elektrische veldvariabelen te berekenen en voeg eventuele bronwaarden toe
4. zolang $n \leq n_{\text{finaal}}$, incrementeer n en ga terug naar stap 2

Dit schema is zeer efficiënt aangezien de matrix \mathbf{G}_{12} een zeer dunbezette matrix is (*E. sparse*) en aangezien de oude waarden niet dienen bijgehouden te worden. In Fig. 5 zijn voor een stuk van een uniform rooster links de veldvariabelen aangeduid die op halve tijdstappen berekend worden, dit zijn de magnetische veldvariabelen, en rechts de veldvariabelen die op gehele tijdstappen berekend worden, dit zijn de elektrische veldvariabelen. Aangezien de berekening van de elektrische veldvariabelen de oude waarde van deze veldvariabelen en de waarde van de magnetische veldvariabelen op een tussengelegen tijdstip vereist, en gelijkaardig voor de berekening van de magnetische veldvariabelen, wordt deze manier van itereren ook aangeduid als zijnde een haasje-over (*E. leapfrog*) iteratieschema.



Figuur 5: Tweemaal hetzelfde stuk van een uniform rooster. Links zijn de veldvariabelen die op halve tijdstappen berekend worden aangeduid, rechts zijn de veldvariabelen die op gehele tijdstappen berekend worden weergegeven.

Met deze expliciete methode is een stabiliteitsvoorwaarde verbonden:

$$\Delta_t \leq \frac{\Delta}{\sqrt{2}c_0} \quad (15)$$

de tijdstap mag niet groter zijn dan een waarde evenredig met de ruimtestap, dit is beter bekend als de Courant voorwaarde [11].

3.2 Subdomein FDTD

In dit werk worden er twee nieuwe iteratieschema's voorgesteld: de subdomein FDTD methode en de veralgemeende subdomein FDTD methode. Beide schema's zijn ook haasje-over iteratieschema's. Het verschil bestaat erin dat het niet enkel gaat om veldvariabelen, bijvoorbeeld elektrisch, maar ook om subdomeinen, bijvoorbeeld E -type, die hand in hand met deze veldvariabelen berekend worden. Eerst wordt de subdomein FDTD methode voorgesteld daarna de veralgemeende subdomein FDTD methode.

3.2.1 De subdomein FDTD methode

Indien het toestandsmodel van een E -type subdomein wordt beschouwd (6), dan weten we dat de uitgangsvariabelen, de variabelen in \mathbf{y} , elektrische veldvariabelen zijn en dat de ingangsvariabelen, de elementen van \mathbf{u} ,

magnetische veldvariabelen zijn. De vector \mathbf{x} bevat zowel elektrische als magnetische veldvariabelen. Gebruiken we nu de benaderingen

$$\dot{\mathbf{x}}|^{n+1/2} \simeq \frac{\mathbf{x}|^{n+1} - \mathbf{x}|^n}{\Delta_t} \quad (16a)$$

$$\mathbf{x}|^{n+1/2} \simeq \frac{\mathbf{x}|^{n+1} + \mathbf{x}|^n}{2} \quad (16b)$$

om de eerste vergelijking van (6) te discretiseren rond $t = (n + 1/2)\Delta_t$, en schrijven we de tweede vergelijking op op het ogenblik $t = (n + 1)\Delta_t$ dan wordt dit:

$$\begin{cases} \mathbf{x}|^{n+1} = \left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2}\right)^{-1} \left(\frac{\mathbf{C}}{\Delta_t} - \frac{\mathbf{G}}{2}\right) \mathbf{x}|^n + \left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2}\right)^{-1} \mathbf{B}\mathbf{u}|^{n+1/2} \\ \mathbf{y}|^{n+1} = \mathbf{L}^T \mathbf{x}|^{n+1} \end{cases} \quad (17)$$

Bemerk dat deze keer de volledige vector \mathbf{x} , dit betekent dat zowel elektrische als magnetische veldvariabelen, benaderd wordt op hetzelfde ogenblik.

Deze iteratievergelijking kan gemakkelijk gebruikt worden in combinatie met de vergelijkingen van de standaard FDTD methode. De ingangsvariabelen, vector \mathbf{u} zijn vereist op $t = (n + 1/2)\Delta_t$. Anderzijds zijn het magnetische veldvariabelen die zich bevinden in een gebied waarvan de veldvariabelen berekend worden met (13). Vergelijking (13a) toont aan dat de waarden ook op die tijdstippen berekend worden. De uitgangsvariabelen, vector \mathbf{y} , zijn elektrische veldvariabelen die worden berekend op tijdstippen $t = (n + 1)\Delta_t$. Deze waarden zijn dus beschikbaar op dezelfde tijdstippen als dat de elektrische veldvariabelen berekend worden in de standaard FDTD techniek. Dit betekent dat elektrische veldvariabelen in een standaard FDTD domein en een E -type subdomein enerzijds en de magnetische veldvariabelen van hetzelfde standaard FDTD domein met een haasje-over techniek kunnen berekend worden.

Dezelfde redenering kan nu toegepast worden voor een H -type subdomein. De ingangsveranderlijken zijn elektrische en de uitgangsveranderlijken zijn magnetische veldvariabelen. Door de eerste vergelijking van (6) te discretiseren rond $t = n\Delta_t$ en de tweede vergelijking op $t = (n + 1/2)\Delta_t$ krijgen we voor een dergelijk subdomein:

$$\begin{cases} \mathbf{x}|^{n+1/2} = \left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2}\right)^{-1} \left(\frac{\mathbf{C}}{\Delta_t} - \frac{\mathbf{G}}{2}\right) \mathbf{x}|^{n-1/2} + \left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2}\right)^{-1} \mathbf{B}\mathbf{u}|^n \\ \mathbf{y}|^{n+1/2} = \mathbf{L}^T \mathbf{x}|^{n+1/2} \end{cases} \quad (18)$$

Subdomeinen van deze soort kunnen berekend worden samen met de magnetische veldvariabelen in een standaard FDTD domein.

Beschouwen we een probleemruimte waarbinnen subdomeinen aangeduid worden die elkaar niet raken. Daarbij dient er ook gezorgd te worden dat ieder subdomein van een specifiek type is: ofwel E -type ofwel H -type. Het overblijvend gebied is dan het gebied waar de standaard FDTD vergelijkingen gebruikt worden. Dan kan het volgende algoritme voorgesteld worden:

1. start met $n = 0$ en zet alle veldwaarden op nul
2. gebruik (18) om de nieuwe waarden binnen de H -type subdomeinen te berekenen en bereken de nieuwe waarden van de magnetische veldvariabelen in het overblijvend gebied met de standaard FDTD vergelijkingen; voeg eventuele brontermen toe
3. gebruik (17) om de nieuwe waarden binnen de E -type subdomeinen te berekenen en bereken de nieuwe waarden van de elektrische veldvariabelen in het overblijvend gebied met de standaard FDTD vergelijkingen; voeg eventuele brontermen toe
4. zolang $n \leq n_{\text{final}}$, incrementeer n en ga terug naar stap 2

Dit wordt het subdomein FDTD algoritme geheten. Verfijning mag binnen subdomeinen toegepast worden.

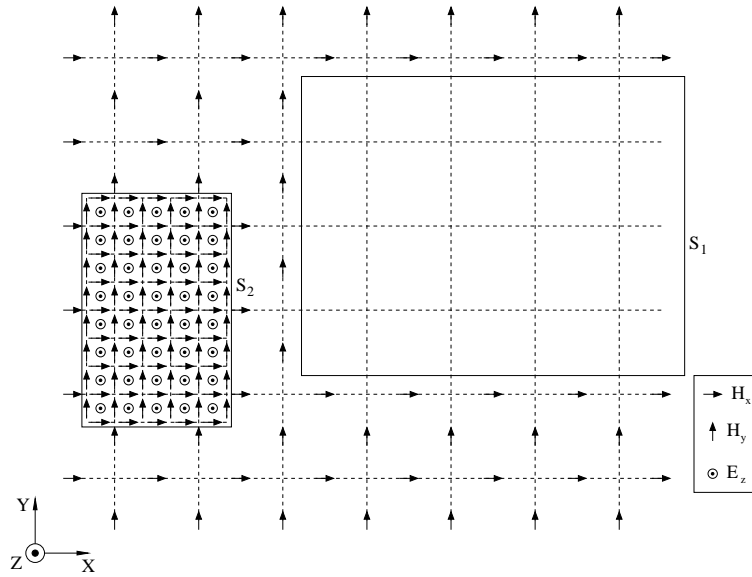
In Fig. 6 en Fig. 7 is tweemaal hetzelfde stuk rooster weergegeven. Het gebied bevat twee subdomeinen: snede S_1 bakent een E -type subdomein af en snede S_2 een H -type subdomein. Binnen het subdomein van S_2 is verfijning toegepast, $r = 3$. Tijdens een tijdstap wordt een nieuwe waarde voor elke veldvariabele eenmaal berekend, dit kan op de halve tijdstap zijn of op de gehele tijdstap. In Fig. 6 zijn de veldvariabelen aangeduid die op $t = (n + \frac{1}{2})\Delta_t$ berekend worden: de H -veldvariabelen in het overblijvende gebied en het H -type subdomein. In Fig. 7 zijn de veldvariabelen aangeduid die op $t = (n + 1)\Delta_t$ berekend worden: de E -veldvariabelen in het overblijvende gebied en het E -type subdomein.

De tijdsdiscretisatie van de subdomeinen brengt echter een kost met zich mee: een aantal matrices die gebruikt worden om de nieuwe waarden te berekenen

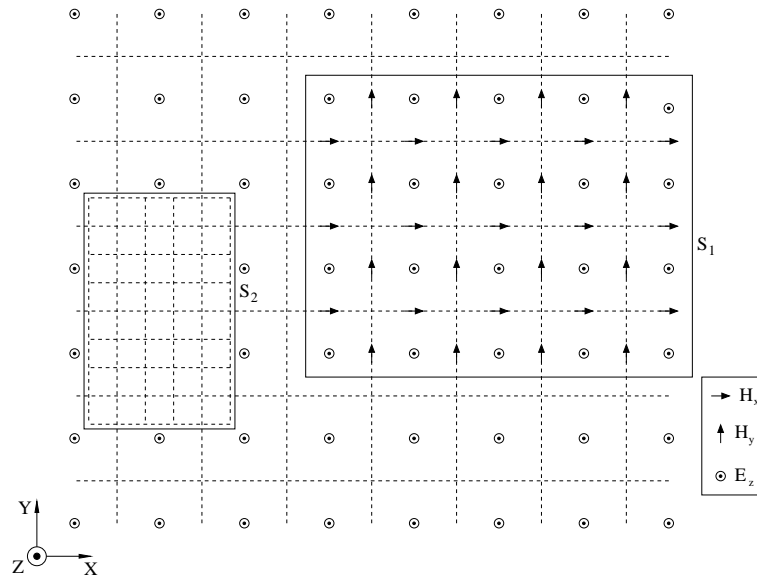
$$\left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2} \right)^{-1} \left(\frac{\mathbf{C}}{\Delta_t} - \frac{\mathbf{G}}{2} \right) \quad (19a)$$

$$\left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2} \right)^{-1} \mathbf{B} \quad (19b)$$

zijn volle matrices aangezien \mathbf{G} dunbezet maar niet diagonaal is. De berekeningen vereist om een tijdstap verder te rekenen zijn daardoor sterk



Figuur 6: De velden die berekend worden op $t = (n + \frac{1}{2})\Delta_t$.



Figuur 7: De velden die berekend worden op $t = n\Delta_t$.

aangegroeid. Een oplossing hiervoor is het gebruik van modellen met gereduceerde orde (*E. reduced order model, ROM*), die op voorhand berekend dienen te worden. Hierop wordt in de volgende sectie dieper ingegaan.

Een tweede luik om het berekenen van de nieuwe waarden van een subdomein te versnellen is door een verandering van variabelen in te voeren zodanig dat de iteratiematrix (19a) een reële matrix blijft maar slechts een minimum aantal niet-nul elementen bevat. Dit kan gebeuren door de reële blok diagonalisatie van (19a) te berekenen:

$$\left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2} \right)^{-1} \left(\frac{\mathbf{C}}{\Delta_t} - \frac{\mathbf{G}}{2} \right) = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^{-1} \quad (20)$$

De verandering van variabelen is dan $\mathbf{z} = \mathbf{P}^{-1} \mathbf{x}$. De iteratievergelijking, b.v. (18), wordt dan:

$$\begin{cases} \mathbf{z}|^{n+1/2} = \mathbf{\Lambda} \mathbf{z}|^{n-1/2} + \mathbf{P}^{-1} \left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2} \right)^{-1} \mathbf{B} \mathbf{u}|^n \\ \mathbf{y}|^{n+1/2} = \mathbf{L}^T \mathbf{P} \mathbf{z}|^{n+1/2} \end{cases} \quad (21)$$

en de kost om de nieuwe waarden van het subdomein te berekenen is niet langer kwadratisch, $\mathcal{O}((\dim[\mathbf{x}])^2)$, maar lineair, $\mathcal{O}(\dim[\mathbf{x}])$ met het aantal variabelen van het systeem, zoals bij de standaard FDTD methode. De kost wordt nu voornamelijk bepaald door een matrix met de omvang van \mathbf{B} and \mathbf{L} , met dimensies $N \times p$ en $N \times m$ respectievelijk. Het is daarom van belang om de subdomeingebieden klein te houden en vooral het aantal ingangsvariabelen en het aantal uitgangsvaariabelen te beperken. Verfijning van de subdomeinen is de sleutel daartoe.

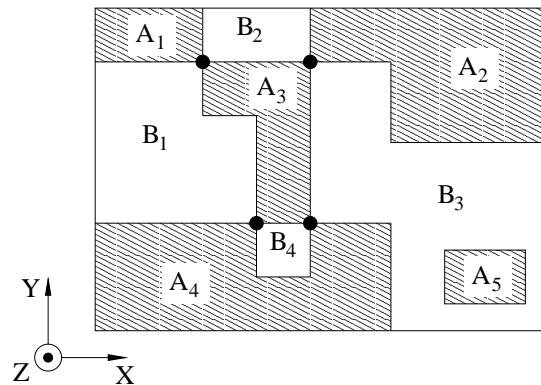
Deze verandering van variabelen heeft geen belang voor wat betreft het iteratieschema aangezien toch begonnen wordt met nulwaarden voor iedere veldvariabele. Wat betreft de implementatie dient op voorhand enkel de reële blokdagonalisatie, m.a.w. $\mathbf{\Lambda}$ en \mathbf{P} , berekend te worden. Met \mathbf{P} wordt dan ook de matrix

$$\mathbf{P}^{-1} \left(\frac{\mathbf{C}}{\Delta_t} + \frac{\mathbf{G}}{2} \right)^{-1} \mathbf{B} \quad (22)$$

berekend. Aangezien eerst een gereduceerd model berekend wordt zijn al deze bewerkingen van toepassing op kleine matrices en zijn ze dus zeker geen beperkende factor.

3.2.2 De veralgemeende subdomein FDTD methode

De veralgemeende subdomein FDTD methode is het tweede nieuwe iteratieschema dat in dit werk voorgesteld wordt. In deze methode wordt de



Figuur 8: Een simulatiedomein opgedeeld in subdomeinen. De subdomeinen kunnen ingekleurd worden met twee kleuren zodanig dat ieder wit subdomein enkel grenst aan grijze subdomeinen en omgekeerd.

probleemruimte volledig opgedeeld in subdomeinen. In tegenstelling tot de subdomein FDTD methode is er, nadat de subdomeinen zijn weggenomen, geen overblijvend domein meer over dat dan met de standaard FDTD methode gediscrètiseerd wordt. Alle subdomeinen samen vormen het volledige simulatiedomein. In Fig. 8 is een voorbeeld van een dergelijk simulatiedomein te zien.

Een vroegere voorwaarde, namelijk dat de veldvariabelen op de rand enkel elektrisch of enkel magnetisch mogen zijn, vervalt. Het is dus toegelaten dat de snede kriskras door het simulatiedomein loopt, en dat zowel elektrische als magnetische veldvariabelen op de rand van hetzelfde subdomein liggen. Vandaar dat deze methode ook de 'veralgemeende' subdomein FDTD methode geheten wordt.

De opdeling in subdomeinen moet echter wel op een zorgvuldige manier gebeuren. Het moet mogelijk zijn de subdomeinen, met gebruik van slechts twee kleuren, b.v. wit en grijs, in te kleuren zodanig dat witte subdomeinen enkel grenzen aan subdomeinen die grijs zijn. Het omgekeerde moet ook gelden: grijze subdomeinen mogen enkel grenzen aan witte subdomein en dus niet aan andere subdomeinen die grijs zijn. In Fig. 8, is een voorbeeld te zien. Een subdomein mag ook volledig binnen een ander subdomein liggen: subdomein A_5 ligt binnen B_3 . In Fig. 8 is de eerste groep subdomeinen, de grijze, voorzien van A -namen: A_1, A_2, \dots, A_5 . De tweede groep subdomeinen, de witte, hebben B -namen gekregen: B_1, B_2, B_3 en B_4 .

Deze opdeling in twee groepen subdomeinen, de A - en de B -gebieden, laat toe om een haasje-over iteratieschema voor te stellen. Terwijl in de

standaard FDTD methode de H - en E -veldvariabelen elkaar afwisselen bij het itereren, wordt dat hier enerzijds de A -subdomeinen en anderzijds de B -subdomeinen. Daartoe dient het toestandsmodel van ieder subdomein gediscretiseerd te worden. Voor de A -subdomeinen gebeurt dit rond het tijdstip $t = n\Delta_t$ en kan men gebruik maken van (18). Voor de B -subdomeinen gebeurt dit rond het tijdstip $t = (n + 1/2)\Delta_t$ en dit resulteert in (17). Dankzij de voorwaarde dat ieder subdomein omringd is door een subdomein van de andere soort, gebeurt de koppeling, uitgedrukt door de vectoren \mathbf{u} en \mathbf{y} , op de geschikte manier.

Het iteratieschema kan dan als volgt geschreven worden:

1. start met $n = 0$ en zet alle veldwaarden op nul
2. gebruik (18) om de nieuwe waarden binnen de A -subdomeinen te berekenen
3. gebruik (17) om de nieuwe waarden binnen de B -subdomeinen te berekenen
4. zolang $n \leq n_{\text{final}}$, incrementeer n en ga terug naar stap 2

Dit is het veralgemeend subdomein FDTD algoritme. De bronnen zijn opgenomen als extra input variabelen in \mathbf{u} in (18) en (17).

Dezelfde twee stappen om het itereren efficiënt te maken worden ook hier gebruikt: eerst wordt een gereduceerd model gegenereerd en vervolgens wordt er van de gereduceerde versie van (19a) de blok diagonalisatie berekend. Deze stappen dienen eenmaal uitgevoerd te worden op voorhand en kunnen dan gebruikt worden in de eigenlijke simulatie.

4 GEREDUCEERDE ORDE MODELLERING (ROM)

4.1 Algemene achtergrond

Recentelijk heeft er vanuit de circuitanalyse een techniek opgang gemaakt om toestandsmodellen te benaderen door kleinere modellen [12–15]. Deze techniek heet gereduceerde orde modellering (*E. reduced order modeling*), en wordt afgekort met ROM. Dikwijls wordt er ook kortweg gesproken van een gereduceerd model. Deze techniek kan ook gebruikt worden om de subdomein FDTD algoritmen efficiënter te laten verlopen.

De techniek bestaat erin, in zekere nader te speciëren zin, het origineel toestandsmodel (6) te vervangen door een nieuw toestandsmodel

$$\begin{cases} \tilde{\mathbf{C}}\dot{\mathbf{z}} = -\tilde{\mathbf{G}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} \\ \mathbf{y} = \tilde{\mathbf{L}}^T\mathbf{z} \end{cases} \quad (23)$$

waarbij de grootte, bepaald door $\dim[\mathbf{z}]$, kleiner is dan de grootte van het originele model $\dim[\mathbf{x}]$. Het aantal interne variabelen, zijnde $\dim[\mathbf{z}]$, is veel kleiner dan $\dim[\mathbf{x}]$. De dimensies van de nieuwe matrices, $\tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{G}}$ en $\tilde{\mathbf{L}}$ zijn overeenkomstig kleiner geworden. De matrices zijn evenwel niet langer dunbezet maar vol. Een gereduceerd model poogt het gedrag van de uitgangsveranderlijken als functie van de ingangsveranderlijken te beschrijven aan de hand van een kleiner aantal interne variabelen.

Het valt te verwachten dat aangezien er veel nullen in de systeemmatrices zitten — ze zijn dunbezet — dat het verband tussen ingangs- en uitgangsveranderlijken relatief eenvoudig is, alleszins eenvoudiger dan het grote aantal interne variabelen ($\dim[\mathbf{x}]$) laat uitschijnen, en dat deze techniek zinvol kan zijn om het verband tussen ingangs- en uitgangsveranderlijken door een veel kleiner systeem te benaderen.

Om te verklaren hoe het reduceren in zijn werk gaat dient de transfermatrix in het Laplace domein voor beide beschouwd te worden:

$$\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{L}^T (\mathbf{G} + s\mathbf{C})^{-1} \mathbf{B} \quad (24a)$$

$$\tilde{\mathbf{H}}(s) = \frac{\tilde{\mathbf{y}}(s)}{\tilde{\mathbf{u}}(s)} = \tilde{\mathbf{L}}^T (\tilde{\mathbf{G}} + s\tilde{\mathbf{C}})^{-1} \tilde{\mathbf{B}} \quad (24b)$$

In een eerste stap wordt een bilineaire transformatie, ook Möbius transformatie genoemd, doorgevoerd:

$$s = \frac{a\sigma + b}{c\sigma + d} \quad \sigma = -\frac{b - sd}{a - sc} \quad (25)$$

waarbij dat $a, b, c, d \in \mathbb{R}$ en de voorwaarde

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} \neq 0 \quad (26)$$

moet gelden. De transfer matrices kunnen dan geschreven worden als:

$$\mathbf{H}(\sigma) = (c\sigma + d)\mathbf{L}^T (\mathbf{I} - \sigma\mathbf{A})^{-1} \mathbf{R} \quad (27a)$$

$$\tilde{\mathbf{H}}(\sigma) = (c\sigma + d)\tilde{\mathbf{L}}^T (\mathbf{I} - \sigma\tilde{\mathbf{A}})^{-1} \tilde{\mathbf{R}} \quad (27b)$$

waarbij \mathbf{A} en \mathbf{R} verbonden zijn met \mathbf{C}, \mathbf{G} en \mathbf{B} :

$$\mathbf{A} = -(d\mathbf{G} + b\mathbf{C})^{-1}(c\mathbf{G} + a\mathbf{C}) \quad (28a)$$

$$\mathbf{R} = (d\mathbf{G} + b\mathbf{C})^{-1}\mathbf{B} \quad (28b)$$

en gelijkaardig voor $\tilde{\mathbf{A}}$ en $\tilde{\mathbf{R}}$. De matrix \mathbf{I} stelt de eenheidsmatrix voor en heeft de dimensie van \mathbf{A} in (27a) en van $\tilde{\mathbf{A}}$ in (27b). In vergelijking (27) speelt de term $(c\sigma + d)$ geen rol meer en het komt er nu op aan ervoor te

zorgen dat het resterende deel van beide vergelijkingen in zekere zin aan elkaar gelijk is.

Deze benadering is gebaseerd op de volgende matrix gelijkheid:

$$(\mathbf{I} - \mathbf{E})^{-1} = \sum_{i=0}^{\infty} \mathbf{E}^i \quad \text{als} \quad \rho(\mathbf{E}) < 1 \quad (29)$$

waarbij $\rho(\mathbf{E})$ de spectrale straal, of de grootste eigenwaarde in absolute waarde van \mathbf{E} , weergeeft. De systeemmatrices worden dan in de buurt van $\sigma = 0$ geëxpandeerd als:

$$\mathbf{H}(\sigma) = (c\sigma + d) \sum_{i=0}^{\infty} \mathbf{L}^T \mathbf{A}^i \mathbf{R} \sigma^i = (c\sigma + d) \sum_{i=0}^{\infty} \mathbf{M}_i \sigma^i \quad (30a)$$

$$\tilde{\mathbf{H}}(\sigma) = (c\sigma + d) \sum_{i=0}^{\infty} \tilde{\mathbf{L}}^T \tilde{\mathbf{A}}^i \tilde{\mathbf{R}} \sigma^i = (c\sigma + d) \sum_{i=0}^{\infty} \tilde{\mathbf{M}}_i \sigma^i \quad (30b)$$

De matrices \mathbf{M}_i en $\tilde{\mathbf{M}}_i$ worden de momenten of Markov parameters genoemd. Het komt er nu op aan ervoor te zorgen dat een zo groot mogelijk aantal momenten van het originele systeem, \mathbf{M}_i gelijk is aan momenten van het gereduceerde systeem $\tilde{\mathbf{M}}_i$:

$$\mathbf{M}_i = \tilde{\mathbf{M}}_i \quad \text{voor} \quad 0 \leq i \leq q \quad (31)$$

Op die manier wordt ervoor gezorgd dat

$$\mathbf{H}(\sigma) = \tilde{\mathbf{H}}(\sigma) + (c\sigma + d)\mathcal{O}(\sigma^q) \quad (32)$$

In deze context is de parameter q van belang, want deze geeft aan in welke mate beide aan elkaar gelijk zijn en wordt de orde van benadering geheten. Hoe groter q , hoe beter het gereduceerde model een benadering is van het originele model.

4.2 Het Laguerre-SVD reductiealgoritme

In dit werk wordt gebruik gemaakt van het Laguerre-SVD reductiealgoritme dat recent binnen de vakgroep ontwikkeld werd [14, 15]. In deze techniek zijn σ en s verbonden door:

$$\sigma = \frac{s - \alpha}{s + \alpha} \quad (33)$$

of de waarden van a, b, c, d zijn:

$$\begin{aligned} a &= \alpha & b &= \alpha \\ c &= -1 & d &= 1 \end{aligned} \quad (34)$$

Door deze verandering van variabelen kan aangetoond worden dat de ontbinding in machten van σ overeenkomt met een ontbinding in geschaalde Laguerre functies [14, 15]. Deze functies zijn orthonormaal zowel in het tijds- als in het frequentiedomein. Een ontbinding in machten van s daarentegen heeft deze eigenschap niet. Deze aanpak is dus theoretisch beter onderbouwd. De matrix \mathbf{A} en \mathbf{R} zijn dan:

$$\mathbf{A} = -(\alpha\mathbf{C} + \mathbf{G})^{-1}(\alpha\mathbf{C} - \mathbf{G}) \quad \mathbf{R} = (\alpha\mathbf{C} + \mathbf{G})^{-1}\mathbf{B} \quad (35)$$

Een volgende stap in de techniek is het invoeren van de Krylov subruimte:

$$\mathcal{K}(\mathbf{A}, \mathbf{R}, q) = \text{span}\{\mathbf{K}\} \quad (36)$$

waarbij \mathbf{K} de Krylovmatrix is:

$$\mathbf{K} = [\mathbf{R}, \mathbf{A}\mathbf{R}, \mathbf{A}^2\mathbf{R}, \dots, \mathbf{A}^{q-1}\mathbf{R}] \quad (37)$$

De dimensie van \mathbf{K} is $pq \times N$, met $p = \dim[\mathbf{u}]$ en $N = \dim[\mathbf{x}]$. Deze subruimte heeft als eigenschap dat ze alle informatie van de q eerste momenten \mathbf{M}_i bevat. Het blijkt nu dat een basis van deze subruimte, \mathbf{U} , kan aangewend worden om het systeem op te projecteren. Dan zijn de nieuwe matrices, $\tilde{\mathbf{C}}, \tilde{\mathbf{G}}, \tilde{\mathbf{B}}, \tilde{\mathbf{L}}$ als volgt gerelateerd met de originele matrices $\mathbf{C}, \mathbf{G}, \mathbf{B}, \mathbf{L}$:

$$\begin{aligned} \tilde{\mathbf{C}} &= \mathbf{U}^T \mathbf{C} \mathbf{U} & \tilde{\mathbf{G}} &= \mathbf{U}^T \mathbf{G} \mathbf{U} \\ \tilde{\mathbf{B}} &= \mathbf{U}^T \mathbf{B} & \tilde{\mathbf{L}} &= \mathbf{U}^T \mathbf{L} \end{aligned} \quad (38)$$

Deze matrices zijn reëel en $\tilde{\mathbf{C}}, \tilde{\mathbf{G}}$ hebben dimensie $pq \times pq$, $\tilde{\mathbf{B}}$ heeft dimensie $pq \times p$ en $\tilde{\mathbf{L}}$ heeft dimensie $pq \times m$. Het kan bewezen worden dat als matrices van het originele systeem en het gereduceerde systeem op die manier verbonden zijn de eerste q momenten van beide systemen gelijk zijn (31). In de hier gebruikte methode, wordt een basis van de Krylov matrix \mathbf{K} berekend aan de hand van de singuliere waarden ontbinding (*E. singular value decomposition, SVD*). Dit algoritme is zeer robuust.

De parameter q is de belangrijkste parameter die in het reductiealgoritme gebruikt wordt: een grotere waarde van q geeft een groter gereduceerd model of meer interne variabelen, $pq = \dim[\mathbf{z}]$, maar geeft aan de andere kant ook aanleiding tot een model dat het gedrag van het origineel beter benadert. Aangezien de dimensie van het gereduceerde systeem evenredig is met het aantal ingangsveranderlijken, is het van belang om het aantal ingangsveranderlijken zo klein mogelijk te houden. Vandaar het belang van verfijning tijdens de spatiale discretisatiestap: het laat toe om veel veldvariabelen binnen het subdomein te koppelen met behulp van een gering aantal ingangsvariabelen met de rest van het simulatiedomein. Bij de voorbeelden zal er ook op gelet worden dat de subdomeinen niet te

veel cellen van het grove rooster beslaan, tenzij in het geval dat het aantal veldvariabelen op de rand op een andere manier kan beperkt worden.

Samengevat omvat het reductiealgoritme de volgende stappen:

1. selecteer q en α
2. los het stelsel $(\mathbf{G} + \alpha\mathbf{C})\mathbf{T}_0 = \mathbf{B}$ op voor \mathbf{T}_0
3. los het stelsel $(\mathbf{G} + \alpha\mathbf{C})\mathbf{T}_k = (\mathbf{G} - \alpha\mathbf{C})\mathbf{T}_{k-1}$, voor $k = 1, 2, \dots, q - 1$ zodanig dat $\mathbf{T}_i = \mathbf{A}^i \mathbf{R}$
4. bouw de Krylov matrix, $\mathbf{K} = [\mathbf{T}_0 \mathbf{T}_1 \dots \mathbf{T}_{q-1}]$, op
5. bereken de singuliere waarden decompositie van \mathbf{K} : $\mathbf{K} = \mathbf{U}\Sigma\mathbf{V}^T$
6. bereken de nieuwe matrices (38)

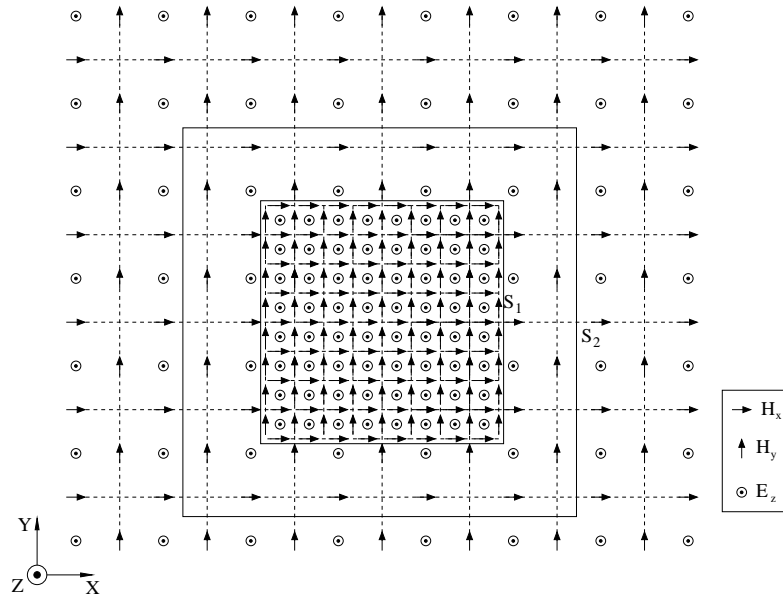
Twee stappen in dit algoritme vragen het meeste rekenwerk. Ten eerste is dat het q maal oplossen van een stelsel van omvang N , en dan vooral de LU-decompositie van $(\mathbf{G} + \alpha\mathbf{C})$. Met gebruik van routines geoptimaliseerd voor dunbezette matrices kan de rekenkost van deze stap in de hand gehouden worden. Ten tweede is dat het berekenen van de SVD van \mathbf{K} , de Krylov matrix.

4.3 Algoritmen

De uiteindelijke algoritmen voor het iteratieschema werden reeds eerder gegeven, maar de stappen die dienen uitgevoerd te worden op voorhand worden hier nog eens op een rijtje gezet.

Gebruik makend van Fig. 9 kunnen de initiële stappen die vereist zijn bij het subdomein FDTD algoritme verduidelijkt worden. Deze zijn de volgende:

1. Bepaal binnen het simulatiedomein de zones waar de velden veel variatie vertonen en verfijn het veld daar. In Fig. 9 is dit het gebied binnen snede S_1 .
2. Schrijf het toestandsmodel voor het subdomein dat zich bevindt binnen S_1 uit.
3. Gebruik het Laguerre-SVD algoritme om het toestandsmodel te benaderen door een toestandsmodel met gereduceerde orde.
4. Indien gewenst kan het gereduceerde toestandsmodel uitgebreid worden met de veldvariabelen die zich tussen snede S_1 en S_2 bevinden. Dan wordt er een toestandsmodel van het subdomein behorend bij



Figuur 9: Een stuk van een typisch simulatiedomein bij de subdomein FDTD methode. Het simulatiedomein bevat één subdomein.

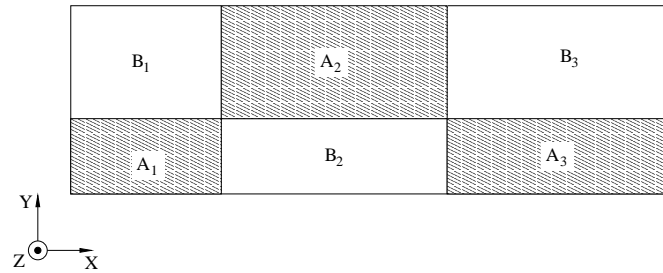
snede S_2 bekomen. De subdomeinen behorend bij S_1 en S_2 in Fig. 9 zijn beide H -type subdomeinen.

5. Discretiseer de tijd voor de subdomeinen. Voor H -type subdomeinen is dit rond $t = n\Delta_t$, vergelijking (18), voor E -type subdomeinen is dit rond $t = (n + \frac{1}{2})\Delta_t$, vergelijking (17).
6. Bereken de reële eigenwaarde decompositie (20) van iedere iteratiematrix en bereken de matrices zoals gebruikt in de iteratievergelijking (21).

Eenmaal dit allemaal gedaan is kan het itereren beginnen.

Gebruik makend van Fig. 10 kunnen de initiële stappen die vereist zijn bij het veralgemeend subdomein FDTD algoritme verduidelijkt worden. Deze zijn de volgende:

1. Beschouw het simulatiedomein en deel het op in een aantal subdomeinen. Om het algoritme efficiënt te laten verlopen dient ervoor gezorgd te worden dat het aantal variabelen aan de grens van ieder subdomein zo klein mogelijk is. Zorg ervoor dat een subdomein van een bepaalde kleur enkel omgeven wordt door subdomeinen met een



Figuur 10: Een typisch simulatiedomein bij de veralgemeende subdomein FDTD methode. De twee groepen subdomeinen zijn aangeduid.

andere kleur. De grijze subdomeinen zijn de A -subdomeinen, de witte subdomeinen zijn de B -subdomeinen.

2. Schrijf het toestandsmodel van ieder subdomein uit.
3. Genereer een gereduceerd model voor elk subdomein.
4. Discretiseer de tijd, voor A -subdomeinen is dit rond $t = n\Delta_t$, vergelijking (18), en voor B -subdomeinen is dit rond $t = (n + 1/2)\Delta_t$, vergelijking (17).
5. Bereken de reële eigenwaarde decompositie (20) van iedere iteratiematrix en bereken de matrices zoals gebruikt in de iteratievergelijking (21).

Eenmaal dit allemaal gedaan is kan het itereren beginnen.

5 STABILITEIT

In de uitgebreide versie van dit werk staat een zeer belangrijke bijdrage over de oorzaken van instabiliteit. Stabiliteit is een belangrijk probleem in de FDTD methode omdat het een expliciete methode is. Daarenboven zijn veel uitbreidingen van de standaard FDTD methode in min of meerdere mate instabiël. Meestal uit zich dit in late-tijd instabiliteit (*E. late-time instability*): resultaten worden vanaf een bepaald ogenblik gedomineerd door een vals exponentieel stijgend resultaat. Aangezien dit onderwerp zeer mathematisch is en dat de resultaten bijdragen tot een beter begrip van de oorzaken, maar in de praktijk weinig gevolg hebben, zullen we in deze samenvatting ons beperken tot twee belangrijke aspecten: het belang

van spatiale reciprociteit enerzijds en de stabiliteit van de subdomein vergelijkingen op zichzelf anderzijds.

5.1 Spatiale reciprociteit

Hiervoor werd uiteengezet dat het toestandsmodel van een simulatiedomein dat gebaseerd is op een uniform rooster en afgesloten is door een perfecte geleider de volgende vorm heeft:

$$\begin{bmatrix} \mathbf{D}_\mu & 0 \\ 0 & \mathbf{D}_\epsilon \end{bmatrix} \begin{bmatrix} \dot{\mathbf{h}} \\ \dot{\mathbf{e}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma_m} & \mathbf{G}_{12} \\ -\mathbf{G}_{12}^T & \mathbf{D}_{\sigma_e} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \mathbf{e} \end{bmatrix} \quad (39)$$

De invulling van \mathbf{G}_{12} speelt hier geen rol, het is enkel van belang dat het andere blok op de nevendiagonaal van \mathbf{G} de matrix $-\mathbf{G}_{12}^T$ is. Bemerken we nu eerst dat de matrix \mathbf{C} kan geschreven worden als:

$$\mathbf{C} = \begin{bmatrix} \mathbf{D}_\mu & 0 \\ 0 & \mathbf{D}_\epsilon \end{bmatrix} = \begin{bmatrix} \mathbf{D}_\mu^{1/2} & 0 \\ 0 & \mathbf{D}_\epsilon^{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{D}_\mu^{1/2} & 0 \\ 0 & \mathbf{D}_\epsilon^{1/2} \end{bmatrix} = \mathbf{C}^{1/2} \mathbf{C}^{1/2} \quad (40)$$

aangezien \mathbf{C} een diagonale matrix is met strikt positief reële getallen. De matrix $\mathbf{C}^{1/2}$ is dan de matrix met op de diagonaal de vierkantswortel van het overeenkomstige diagonaalelement van \mathbf{C} . Daar de elementen van \mathbf{C} strikt positief zijn bestaat ook de matrix $\mathbf{C}^{-1/2}$. Door gebruik te maken hiervan is het mogelijk om (39) te herschrijven als

$$\dot{\mathbf{x}}' = \begin{bmatrix} \dot{\mathbf{h}}' \\ \dot{\mathbf{e}}' \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_\mu^{-1/2} \mathbf{D}_{\sigma_m} \mathbf{D}_\mu^{-1/2} & \mathbf{D}_\epsilon^{-1/2} \mathbf{G}_{12} \mathbf{D}_\mu^{-1/2} \\ -\mathbf{D}_\mu^{-1/2} \mathbf{G}_{12}^T \mathbf{D}_\epsilon^{-1/2} & \mathbf{D}_\epsilon^{-1/2} \mathbf{D}_{\sigma_e} \mathbf{D}_\epsilon^{-1/2} \end{bmatrix} \begin{bmatrix} \mathbf{h}' \\ \mathbf{e}' \end{bmatrix} = -\mathbf{G}' \mathbf{x}' \quad (41)$$

waarbij $\mathbf{h}' = \mathbf{D}_\mu^{1/2} \mathbf{h}$ en $\mathbf{e}' = \mathbf{D}_\epsilon^{1/2} \mathbf{e}$.

De eigenwaarden van \mathbf{G}' bepalen volledig de polen van het systeem: als er geen eigenwaarden van \mathbf{G}' in het linkerhalfvlak liggen, dan liggen er geen polen van het systeem in het rechterhalfvlak. De blokken van \mathbf{G}' hebben de volgende eigenschappen: de blokken op de diagonaal, \mathbf{G}'_{11} en \mathbf{G}'_{22} , zijn diagonaal matrices met waarden die positief zijn. De blokken op de nevendiagonaal zijn elkaars tegengestelde:

$$\mathbf{G}'_{12}{}^T = (\mathbf{D}_\epsilon^{-1/2} \mathbf{G}_{12} \mathbf{D}_\mu^{-1/2})^T = \mathbf{D}_\mu^{-1/2} \mathbf{G}_{12}^T \mathbf{D}_\epsilon^{-1/2} = -\mathbf{G}'_{21}{}^T \quad (42)$$

Als een matrix \mathbf{K} gesplitst wordt, enerzijds in \mathbf{S} , zijn symmetrisch deel, en anderzijds in \mathbf{A} , zijn anti-symmetrisch deel, dan is het zo dat de eigenwaarden van het symmetrisch deel allemaal reëel zijn en die van het

anti-symmetrische deel puur imaginair. Voor de eigenwaarden, λ_j , van \mathbf{K} geldt er dan [16]:

$$\min[\lambda(\mathbf{S})] \leq \operatorname{Re}[\lambda_j] \leq \max[\lambda(\mathbf{S})] \quad \forall j \quad (43)$$

$$\min[\lambda(\mathbf{A})/i] \leq \operatorname{Im}[\lambda_j] \leq \max[\lambda(\mathbf{A})/i] \quad \forall j \quad (44)$$

waarbij $i = \sqrt{-1}$, $\lambda(\mathbf{S})$ de eigenwaarden van \mathbf{S} en $\lambda(\mathbf{A})$ de eigenwaarden van \mathbf{A} bevat. Dit betekent met andere woorden dat een rechthoekig gebied in het imaginaire vlak kan afgebakend worden waarbinnen de eigenwaarden van \mathbf{K} gelegen zijn. De linker grens van dit gebied is de kleinste eigenwaarde van \mathbf{S} , de rechter grens is de grootste eigenwaarde van \mathbf{S} en de bovengrens en ondergrens van dit gebied wordt bepaald door de meest extreme waarden van \mathbf{A} .

Passen we dit toe op \mathbf{G}' , dan merken we eerst op dat onze interesse enkel uitgaat naar de linker- en rechtergrens van het gebied waarbinnen alle eigenwaarden gelegen zijn, en deze grenzen worden bepaald door het symmetrisch deel van \mathbf{G}' . Het symmetrisch deel van \mathbf{G}' is:

$$\frac{\mathbf{G}' + \mathbf{G}'^T}{2} = \begin{bmatrix} \mathbf{D}_\mu^{-1/2} \mathbf{D}_{\sigma_m} \mathbf{D}_\mu^{-1/2} & 0 \\ 0 & \mathbf{D}_\epsilon^{-1/2} \mathbf{D}_{\sigma_e} \mathbf{D}_\epsilon^{-1/2} \end{bmatrix} \quad (45)$$

een diagonaalmatrix die enkel positief reële waarden bevat. Dit betekent dat het reële deel van geen enkele eigenwaarde van \mathbf{G}' negatief is. Anders gezegd geen enkele pool van een reciprook systeem bevindt zich in het rechter halfvlak. Dit toont het belang aan van de spatiale reciprociteit. Bij systemen die niet gebaseerd zijn op een rooster dat spatiaal reciprook is, is het niet langer mogelijk garanties in verband met de ligging van de polen te geven.

5.2 Subdomein vergelijkingen

In deze sectie zullen we aantonen dat de subdomeinvergelijkingen (17) en (18) op zichzelf beschouwd onvoorwaardelijk stabiel zijn, als het toestandsmodel waarvan vertrokken wordt stabiel is. Met andere woorden de eigenwaarden van (19a) zijn dan in absolute waarde niet groter dan 1. Bemerk dat de stabiliteit van een toestandsmodel verbonden is met de polen van een systeem, met de linker- en rechterhalfvlak van het imaginair vlak, en dat de stabiliteit van een tijdsgediscretiseerd systeem, (17) en (18), verbonden is met de eenheidscirkel.

Een eigenwaarde λ van (19a) is ook een eigenwaarde van:

$$\left(\frac{\mathbf{I}}{\Delta_t} + \frac{\mathbf{C}^{-1}\mathbf{G}}{2} \right)^{-1} \left(\frac{\mathbf{I}}{\Delta_t} - \frac{\mathbf{C}^{-1}\mathbf{G}}{2} \right) \quad (46)$$

Indien de matrix $\mathbf{C}^{-1}\mathbf{G}$ gelijkvormig is met de matrix Λ :

$$\mathbf{C}^{-1}\mathbf{G} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1} \quad (47)$$

dan is (46) gelijkvormig met:

$$\left(\frac{\mathbf{I}}{\Delta_t} + \frac{\mathbf{\Lambda}}{2} \right)^{-1} \left(\frac{\mathbf{I}}{\Delta_t} - \frac{\mathbf{\Lambda}}{2} \right) \quad (48)$$

Bijgevolg zijn de eigenwaarden van $\mathbf{C}^{-1}\mathbf{G}$, λ , verbonden met de eigenwaarden van (19a), y , door:

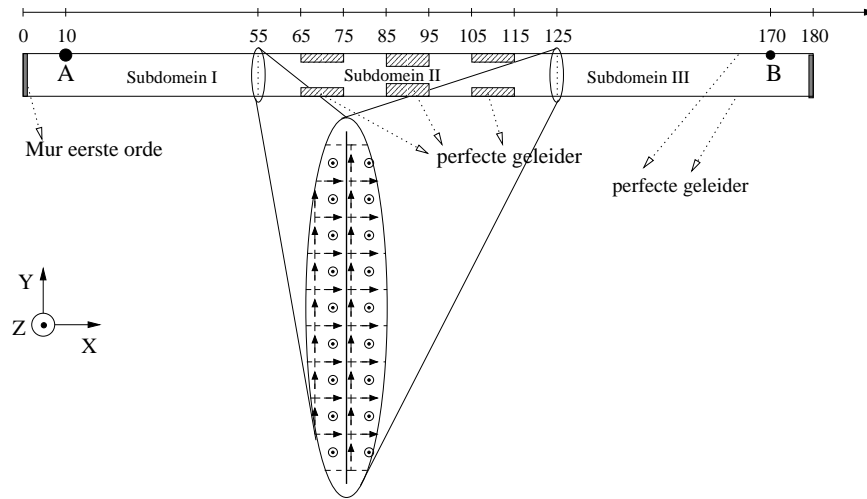
$$y = \frac{1 - \frac{\Delta_t}{2}\lambda}{1 + \frac{\Delta_t}{2}\lambda} \quad (49)$$

Gebruik makend van $\text{Re}(\lambda) \leq 0$, kan er dan aangetoond worden dat $|y| \leq 1$.

We kunnen dan besluiten dat, onafhankelijk van de gekozen tijdstap, de subdomein iteratievergelijking, los van de rest van het simulatiedomein beschouwd, onvoorwaardelijk stabiel is. Het valt dus te verwachten dat bij de combinatie van een fijn rooster en een grof rooster, waarbij het fijn rooster op deze manier gediscretiseerd wordt en het grof rooster met vergelijkingen uit de standaard FDTD methode, de uiteindelijke tijdstap zal liggen in de buurt van de tijdstap geassocieerd met het grof rooster. Dit blijkt ook uit de voorbeelden: de tijdstap kon dikwijls gelijk aan ongeveer de helft van de tijdstap van het grof rooster gekozen worden. Dit is in elk geval veel groter dan de tijdstap geassocieerd met het fijn rooster, in onze voorbeelden vlug vijf tot zes maal groter. De verklaring dat de tijdstap toch een stuk kleiner is dan de tijdstap van het grof rooster heeft te maken met het verlies van spatiale reciprociteit bij de voorgestelde manier om het rooster te verfijnen. We benadrukken dat dit nadeel niet opweegt tegen de nauwkeurigheid die deze aanpak oplevert en die hier prioritair was.

6 NUMERIEKE VOORBEELDEN

Een aantal numerieke voorbeelden zijn uitgewerkt om de twee nieuwe algoritmen te valideren. Voor elk algoritme wordt in deze samenvatting één voorbeeld getoond. Het voorbeeld van de subdomein FDTD methode illustreert duidelijk dat deze methode kan beschouwd worden als een methode om op een meer algemene manier subcel modellen voor algemene geometriën te genereren.



Figuur 11: TEM simulatie probleem.

6.1 Veralgemeende subdomein FDTD methode

In Fig. 11 wordt een TE-probleem voorgesteld, dit is het duale geval van de hiervoor gegeven afleidingen: de betrokken veldcomponenten zijn E_x , E_y en H_z . Deze resultaten werden reeds voorgesteld in [17] en [18]. De reflectie en transmissie van een propagerende TEM golf ten gevolge van een obstakel wordt onderzocht. Een oneindige parallelle plaat golfgeleider wordt gemodelleerd door een 180 cellen lange golfgeleider afgesloten met Mur eerste orde absorberende randvoorwaarden. De ruimtestap is $\Delta = 3,5\text{cm}$ en de afstand tussen beide platen is 9 cellen. Het rooster wordt niet lokaal verfijnd. In het midden van de golfgeleider zijn drie paar perfect geleidende obstakels aangebracht. De locatie, in aantal cellen, is aangeduid op de as. De TEM-mode is de enige propagerende mode in het beschouwde frequentiedomein en de invallende mode wordt geëxciteerd door een stroom geïnjecteerd op $x = 10\Delta$ (locatie A in Fig. 11). De amplitude van de gereflecteerde mode wordt op punt A bekeken, de transmissie van de mode wordt bekeken ter hoogte van punt B.

Het simulatiedomein wordt opgedeeld in drie subdomeinen, een eerste snede bevindt zich op $x = 55\Delta$ en een tweede op $x = 125\Delta$. De snede is zodanig aangebracht dat de veldvariabelen links van de snede H_z componenten zijn en deze rechts E_y componenten zijn. Daardoor is ieder subdomein van dezelfde algemene soort: indien er veldvariabelen op de linkse rand zijn, zijn het E_y veldvariabelen, indien er veldvariabelen op

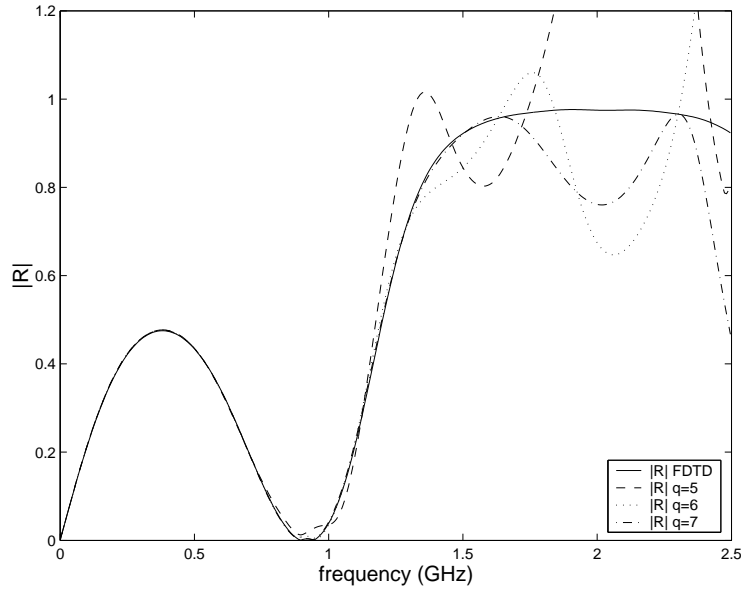
de rechtse rand zijn, zijn het H_z veldvariabelen. Subdomein I bevat 1430 veldveranderlijken, subdomein II 1386 en subdomein III 1439.

Als de snede ver genoeg van de obstakels gekozen wordt, valt het te verwachten dat alle andere niet-propagerende modes uitgestorven zijn eenmaal ze in de buurt van de snede komen. Daarom worden de uitgangsveranderlijken aan elke kant van ieder subdomein vervangen door hun gemiddelde waarde, wat overeenkomt met de amplitude van de TEM-mode. Dit zorgt ervoor dat de simulaties veel sneller verlopen aangezien de grootte van ieder gereduceerd model evenredig is met het aantal ingangsveranderlijken. Het aantal ingangs- en uitgangsveranderlijken van ieder subdomein is twee, $p = 2$:

- voor subdomein I, is dit één ingangs- en uitgangsveranderlijke aan de rechterkant van het subdomein en één ingangsveranderlijke voor de bron in punt A en één uitgangsveranderlijke voor de mode ter hoogte van punt A
- voor subdomein II, is dit één ingangs- en uitgangsveranderlijke aan de linkerkant van het subdomein en één ingangs- en uitgangsveranderlijke aan de rechterkant van het subdomein
- voor subdomein III, is dit één ingangs- en uitgangsveranderlijke aan de linkerkant van het subdomein en één uitgangsveranderlijke voor de mode ter hoogte van punt B en één ingangsveranderlijke voor mogelijk gebruik als bron in punt B .

Een gereduceerd model van ieder subdomein werd gegenereerd. Telkens werd de orde van approximatie voor ieder subdomein gelijk gekozen, $q = q_I = q_{II} = q_{III}$, daardoor is de dimensie van ieder gereduceerd model gelijk: $2q$. Het haasje-over iteratieschema voor de simulatie is hier eenvoudig: op $t = n\Delta_t$ worden de nieuwe waarden voor subdomein I en subdomein III berekend, op $t = (n + \frac{1}{2})\Delta_t$ worden de nieuwe waarden voor subdomein II berekend. De tijdstap werd overeenkomstig de tijdstap bij de standaard FDTD methode gekozen: $\Delta_t = 8,24 \times 10^{-12}$ s.

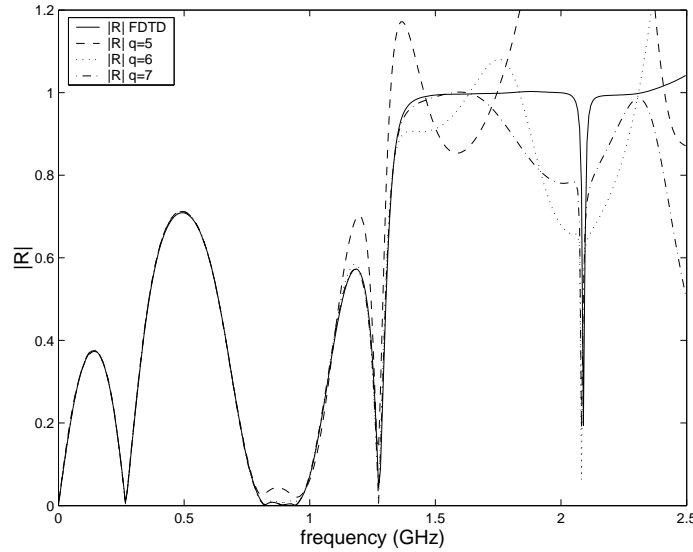
In Fig. 12 is de amplitude van de reflectiecoëfficiënt gegeven voor de simulaties met $q = 5, 6, 7$. Als referentieresultaat wordt het resultaat van de standaard FDTD methode beschouwd. Het is duidelijk te zien dat naarmate de orde van benadering, q , groter wordt de resultaten overeenstemmen voor een breder frequentiegebied. Voor $q = 5$ stemmen de resultaten overeen tot $f = 0,75$ GHz, voor $q = 6$ is dit reeds $f = 1,25$ GHz en voor $q = 7$ is dit $f = 1,5$ GHz. Voor de transmissiecoëfficiënt gaan dezelfde opmerkingen op. Voor hogere q -waarden blijft de maximum frequentie tot waar de simulaties goed zijn stijgen, maar wel aan een lager tempo.



Figuur 12: Amplitude van de reflectiecoëfficiënt voor $q = 5, 6, 7$ vergeleken met het standaard FDTD referentie resultaat.

Omdat de subdomeinen van een algemeen type zijn, op de linkse rand van een subdomein zijn de velden altijd E_y componenten en op de rechtse rand altijd H_z componenten, is het mogelijk om een gereduceerd model meermaals te gebruiken. Om dit te illustreren werd in een simulatie subdomein II tweemaal gebruikt door in Fig. 11 tussen subdomein II en subdomein III nogmaals subdomein II in te lassen. Het haasje-over iteratieschema is dan: op $t = n\Delta_t$ worden de nieuwe waarden van subdomein I en de tweede versie van subdomein II berekend en op $t = (n + \frac{1}{2})\Delta_t$ worden de nieuwe waarden van de eerste versie van subdomein II en subdomein III berekend. In Fig 13 wordt de reflectiecoëfficiënt voor $q = 5, 6, 7$ vergeleken met het referentieresultaat gegenereerd met de standaard FDTD methode. Dezelfde conclusies kunnen getrokken worden als voorheen. Op deze manier is aangetoond dat de subdomeinen en hun gereduceerde versie hergebruikt kunnen worden, hetgeen aanleiding geeft tot een efficiënter algoritme.

De rekentijd voor dit voorbeeld was tussen een factor 25, voor $q = 10$, en 50, voor $q = 5$ maal kleiner dan de standaard FDTD simulatie. De verbetering kon nog opgevoerd worden omdat de tijdstap in de veralgemeende subdomein FDTD methode groter kon gekozen worden. De tijdstap kon,



Figuur 13: Reflectiecoëfficiënt voor $q = 5, 6, 7$, vergeleken met het standaard FDTD referentie resultaat. In het geval dat in het simulatiedomein subdomein II tweemaal gebruikt werd.

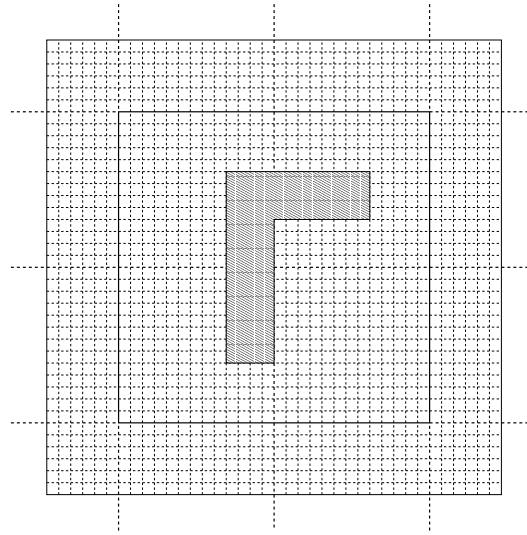
voor $q = 5$, acht maal groter gekozen worden dan de limiet van de standaard FDTD methode. Voor $q = 8$ was dit meer dan vier maal.

In deze gevallen werd geen rekening gehouden met de rekentijd die vereist is in de verschillende stappen die op voorhand dienen uitgevoerd te worden. Bij de simulaties die overeenkomen met de resultaten van Fig. 12 was de simulatietijd driemaal groter dan de tijd die nodig was op voorhand. De berekingen met de nieuwe methode zijn dan nog altijd veel vlugger dan deze met de standaard FDTD methode. De vergelijking van reketijden is natuurlijk bijzonder probleemafhankelijk.

6.2 De subdomein FDTD methode

Omdat de subdomein FDTD methode geschikt is om subcellen te genereren worden hier zowel de termen subcel model als subdomein door elkaar gebruikt. Dit voorbeeld werd samen met andere gepubliceerd in [19]. Andere voorbeelden van deze methode zijn ook verschenen in [20], en het laat ook toe om simulaties van fotonische kristal structuren sneller uit te voeren [21, 22].

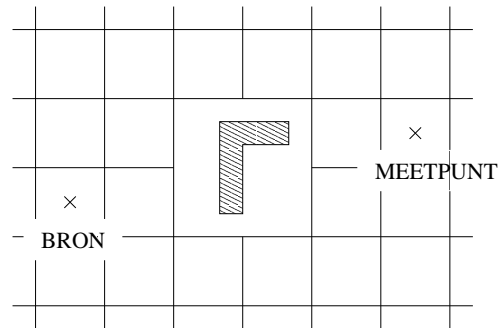
Om de veelzijdigheid van de subdomein FDTD methode aan te tonen



Figuur 14: Het subdomein waarop het gegenereerde subcel model van een verlieshebbend diëlektrisch L-vormig object gebaseerd is, $r = 13$.

wordt een subcel model van een asymmetrisch verlieshebbend object gegenereerd. Dit voorbeeld is een TM-probleem: enkel de veldcomponenten H_x , H_y en E_z zijn aanwezig. Het object heeft een L-vorm en is geplaatst in vacuüm. De elektrische parameters van het materiaal zijn: $\epsilon_r = 10$ en $\sigma_e = 20$ S/m. In Fig. 14 is het fijne rooster getoond, waarop het subcel model gebaseerd is. De verfijningsverhouding is $r = 13$ en de ruimtestappen in beide roosters zijn: $\Delta_f = 0,1$ mm en $\Delta_g = 1,3$ mm. Het subdomein is het fijne rooster en beslaat een oppervlakte van 2×2 grove cellen met daarrond een laag van $(r - 1)/2 = 6$ fijne cellen. Hetzelfde subdomein is gebruikt tijdens het discretiseren van de tijd. Het aantal veldvariabelen in het subdomein of het aantal variabelen in het originele toestandsmodel is $\dim[\mathbf{x}] = 4408$. Het aantal ingangsvariabelen is $\dim[\mathbf{u}] = 12$ en het aantal uitgangsvariabelen is $\dim[\mathbf{y}] = 8$. De dimensie van het subcel model is dan $\dim[\mathbf{z}] = 12q$. Lineaire interpolatie werd gebruikt om de veldvariabelen op de rand van het fijn rooster te koppelen met de veldvariabelen in het grove rooster.

In Fig. 15 wordt de configuratie getoond die gebruikt werd om het model te testen. Een lijnbron aan de ene kant van het subcel model werd geëxciteerd met een puls en aan de andere kant van het subcel model werd het elektrisch veld gemeten. De tijdstap die gebruikt werd in de simulatie was $1,4 \times 10^{-12}$, of ongeveer 0,45 maal de maximum tijdstap verbonden

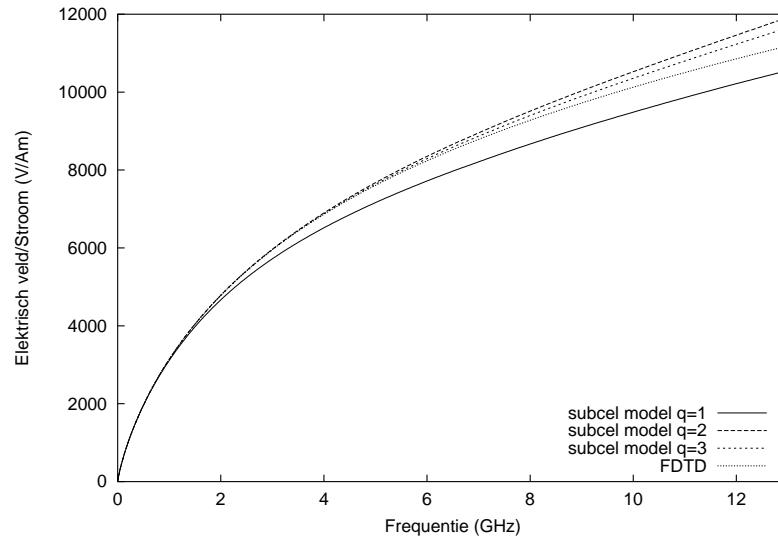


Figuur 15: De testconfiguratie waarmee het L-vormig subcel model getest werd.

aan het grove rooster. De verhouding van beide fouriergetransformeerde resultaten is te zien in Fig. 16. De resultaten kunnen vergeleken worden met de resultaten afkomstig van een standaard FDTD simulatie gebaseerd op het fijn rooster gebruikt doorheen het ganse simulatiedomein. Men kan zien dat naarmate de orde van benadering, q , groter wordt, de hoogste frequentie waarop het subcel model nog geldig is ook groter wordt. Voor $q = 1$ is dit tot 1,5 GHz, voor $q = 2$ is dit reeds 4,5 GHz en voor $q = 3$ is het subcel model geldig tot 6 GHz.

In tabel 1 wordt een vergelijking van de rekentijden, voor dit specifiek voorbeeld tussen nieuwe methode en standaard FDTD methode gemaakt. De simulaties gebaseerd op de subdomein FDTD methode, in functie van q worden vergeleken met de simulatie voor het referentieresultaat dat gegenereerd werd door het fijn rooster doorheen de ganse simulatieruimte te gebruiken. In tabel 1 is de kost van vermenigvuldigingen en delingen enerzijds en van optellingen en aftrekkingen anderzijds opgesplitst. Het aantal bewerkingen vereist voor de standaard FDTD vergelijkingen is opgesplitst volgens de veldcomponenten, $H_x + H_y + E_z$. Het verschil in tijdstap is ook in rekening gebracht, de tijdstap voor de subdomein FDTD methode is ongeveer de helft van de tijdstap normaal te verwachten bij het grof rooster. De tijdstap in de FDTD simulatie voor het referentieresultaat is evenveel kleiner als dat de ruimtestap kleiner is: 13. De efficiëntie komt duidelijk naar voren in deze tabel. De nieuwe aanpak is, zelfs in het geval $q = 3$, een factor 180 tot 460 maal sneller.

In tabel 2 worden de geheugenvereisten, in aantal op te slaan getallen, voor de subdomein FDTD methode vergeleken met die voor de standaard FDTD. Het blijkt dat de geheugenvereisten voor de nieuwe methode 30-maal minder zijn. Daarenboven is het zo dat, als in een simulatie hetzelfde



Figuur 16: Verhouding van het gemeten elektrisch veld tot de stroom in de lijnbron: frequentieresultaten.

model meerdere malen gebruikt wordt, de besparingen nog groter zijn. Reden daarvoor is dat de systeemmatrices van het subcel model slechts eenmaal dienen opgeslagen te worden. Elk extra gebruik van hetzelfde subcel model heeft enkel tot gevolg dat een extra vector dient opgeslagen te worden. In het voorbeeld van de fotonische kristal structuur [21, 22] leverde dit groot voordeel op aangezien daar voor iedere diëlektrische staaf in de periodische structuur hetzelfde model gebruikt werd. In die simulaties wordt hetzelfde model tot 2000 maal gebruikt.

7 BESLUIT

In dit werk worden twee nieuwe FDTD methodes voorgesteld: de subdomein FDTD methode en de veralgemeende subdomein FDTD methode. Beide kunnen als veralgemeningen van de standaard FDTD methode beschouwd worden waarbij subdomeinen op een gelijkaardige manier, met de karakteristieke haasje-over manier van itereren, als de veldvariabelen kunnen behandeld worden. Vooral de subdomein FDTD methode is van belang aangezien deze, gecombineerd met roosterverfijningstechnieken, resulteert in een methode die toelaat om op een algemene manier subcel modellen te genereren.

Tabel 1: Vergelijking: rekenkost.

| | subdomein FDTD | fijn rooster | |
|-------------------------------------------|----------------------------------|-----------------------------------------------------------|------------------|
| kost FDTD vgl. per tijdstap | $48 + 48 + 80$ $24 + 24 + 20$ | $8\,268 + 8\,216 + 16\,224$ $4\,134 + 4\,108 + 4\,152$ | + en - × en / |
| kost vgl. (21) per tijdstap | $12q\,21 - 8$ $12q\,22$ | 0 0 | + en - × en / |
| $\Delta_{t,g}$ per tijdstap | ± 2 | 13 | |
| totaal / $\Delta_{t,g}$, voor $q = 3$ | 924 860 | 425 204 161 122 | + en - × en / |

Tabel 2: Vergelijking: geheugenvereisten.

| | subdomein FDTD | fijn rooster |
|---------------------|----------------|----------------------------|
| FDTD vergelijkingen | $48 + 48 + 40$ | $8\,268 + 8\,216 + 8\,208$ |
| vergelijking (21) | $12q\,23$ | 0 |
| totaal voor $q = 3$ | 828 | 24 692 |

Om de methode efficiënt te maken zijn twee stappen nodig, een eerste het genereren van een gereduceerd model gebaseerd op een nieuwe mathematische techniek. Het tweede, het berekenen van een reële blokdiagonalisatie. Voorbeelden hebben de efficiëntie aangetoond.

De resultaten van dit onderzoek hebben geresulteerd in twee publicaties, als eerste auteur, in internationale tijdschriften [17, 19], vijf artikels in de proceedings van internationale conferenties [18, 20, 21, 23–25] en één abstract in een internationale conferentie [22]. Ten slotte was er één artikel in de proceedings van een nationale conferentie [26].

BIBLIOGRAFIE

- [1] J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 1993.
- [2] R. F. Harrington, *Field Computations by Moment Methods*, Macmillan, New York, 1968.
- [3] A. Taflové and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method, second ed.*, Artech House, 2000.

-
- [4] M. Okoniewski, E. Okoniewska, and M. A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 3, pp. 422-429, Mar. 1997.
 - [5] D. T. Prescott and N. V. Shuley, "A method for incorporating different sized cells into the finite-difference time-domain analysis technique," *IEEE Microwave and Guided Wave Letters*, vol. 2, no. 11, pp. 434-436, Nov. 1992.
 - [6] K. R. Umashankar, A. Taflove, and B. Beker, "Calculation and experimental validation of induced currents on coupled wires in an arbitrary shaped cavity," *IEEE Trans. Antennas Propagat.*, vol. 35, no. 11, pp. 1248-1257, Nov. 1987.
 - [7] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302-307, May 1966.
 - [8] C. J. Railton, I. J. Craddock, and J. B. Schneider, "Improved locally distorted CPFDTD algorithm with provable stability," *Electronics Letters*, vol. 31, no. 18, pp. 1585-1586, Aug. 1995.
 - [9] P. Thoma and T. Weiland, "A consistent subgridding scheme for the finite difference time domain method," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 9, no. 5, pp. 359-374, 1996.
 - [10] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 1995.
 - [11] A. Taflove and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 23, no. 8, pp. 623-630, Aug. 1975.
 - [12] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. on Computer-Aided Design*, vol. 14, no. 5, pp. 639-649, May 1995.
 - [13] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on Computer-Aided Design*, vol. 17, no. 8, pp. 645-654, Aug. 1998.
 - [14] L. Knockaert and D. De Zutter, "Passive reduced order multiport modeling: The Padé-Laguerre, Krylov-Arnoldi-SVD connection," *AEU Int. J. Electron Commun.*, vol. 53, no. 5, pp. 254-260, 1999.

-
- [15] L. Knockaert and D. De Zutter, "Laguerre-SVD reduced-order modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1469–1475, Sep. 2000.
 - [16] A. S. Householder, *The theory of matrices in numerical analysis*, Dover Publications, Inc, New York, 1964.
 - [17] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," *IEEE Microwave and Guided Wave Letters*, vol. 10, no. 8, pp. 301–303, Aug. 2000.
 - [18] B. Denecker, D. De Zutter, L. Knockaert, and F. Olyslager, "A higher level algorithm for 2D electromagnetic modelling using an FDTD grid," *IEEE Antennas Propagat. Symp.*, vol. 3, pp. 1340–1343, July 2000.
 - [19] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Generation of FDTD subcell equations by means of reduced order modeling," *IEEE Trans. Antennas Propagat.*, accepted for publication.
 - [20] B. Denecker, F. Olyslager, D. De Zutter, and L. Knockaert, "2-D FDTD subgridding based on subdomain generation," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 288–290, May 2001.
 - [21] B. Denecker, F. Olyslager, D. De Zutter, L. Klinkenbusch, and L. Knockaert, "Efficient analysis of photonic crystal structures using a novel FDTD-technique," *IEEE Antennas Propagat. Symp.*, vol. 4, pp. 344–347, June 2002.
 - [22] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Efficient FDTD analysis of very large finite photonic crystal structures," *LEOS Benelux, Photonic Crystal Workshop, Ghent*, May 2002.
 - [23] L. Knockaert and B. Denecker, "Explicit reciprocity and reduced order modeling," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 497–499, May 2001.
 - [24] L. Knockaert, B. Denecker, and D. De Zutter, "Explicitly reciprocal reduced order modeling: Laguerre-SVD versus balanced realizations," *IEEE Antennas Propagat. Symp.*, vol. 2, pp. 556–558, June 2002.
 - [25] L. Knockaert, B. Denecker, D. De Zutter, and F. Olyslager, "Reduced order multiport modelling via Laguerre-SVD and its application to FDTD," *XXVIIIth URSI General Assembly, Maastricht, The Netherlands*, pp. CD-ROM, Aug. 2002.

- [26] B. Denecker, "The finite-difference time-domain method and reduced order modelling: Electromagnetics," *1e Doctoraatssymposium FTW, Gent*, p. 9, Dec. 2000.

English Text

CHAPTER 1

Introduction

1 MAXWELL'S EQUATIONS

Electromagnetism is based on Maxwell's equations, published by James C. Maxwell from 1860 on in some papers and later in 1873 in his fundamental work '*A treatise on electricity and magnetism*'. That achievement is still recognized up to now, as indicates this quotation from the Encyclopædia Britannica:

Maxwell is regarded by most modern physicists as the scientist of the 19th century who had the greatest influence on 20th-century physics, and he is ranked with Sir Isaac Newton and Albert Einstein for the fundamental nature of his contributions. In 1931, on the 100th anniversary of Maxwell's birth, Einstein described the change in the conception of reality in physics that resulted from Maxwell's work as 'the most profound and the most fruitful that physics has experienced since the time of Newton'.

In differential form these equations are:

$$\begin{aligned}\nabla \times \mathbf{E}(\mathbf{r}, t) &= -\frac{\partial \mathbf{B}(\mathbf{r}, t)}{\partial t} \\ \nabla \times \mathbf{H}(\mathbf{r}, t) &= \frac{\partial \mathbf{D}(\mathbf{r}, t)}{\partial t} + \mathbf{J}(\mathbf{r}, t) \\ \nabla \cdot \mathbf{B}(\mathbf{r}, t) &= 0 \\ \nabla \cdot \mathbf{D}(\mathbf{r}, t) &= \rho(\mathbf{r}, t)\end{aligned}$$

Maxwell's equations, here in their most compact form, are a very short mathematical description, which explain all electric and magnetic phenomena around us. However, although being very compact, solving and un-

derstanding these equations is not straightforward: ask most, if not all, students who need to study electromagnetism.

Although Maxwell's equations are more than a century old, they still form an important area of research. The main reason is that all signal processing and as a result all telecommunication and computer industry applications, rely on electromagnetism. The design of antennas, used in wireless applications and modeling of interconnect structures in high speed electronics are but two of a long list of examples.

In a recent article [1], the current importance of electromagnetism is discussed. Whereas initially advances in electromagnetics were mainly stimulated by the defense industry, e.g. the application of radar technology urged for the development of microwave technology, the principal drive nowadays comes from the computer and telecommunications industry. Since electronics are being operated at higher frequencies, parasitic effects, which result in electromagnetic interference (EMI), become more and more important. Consider for example an interconnect. At low frequencies this interconnect allows the propagation of a signal from one point to another without disturbing other neighbouring wires. High frequency signals however, propagating along this interconnect, will be a source of crosstalk, to name but one parasitic effect. This crosstalk results in a spurious signal in neighbouring wires. A large spurious signal can result in a malfunctioning device. In [1] this issue is formulated as follows:

The bedrock of introductory circuit analysis, Kirchoff's current and voltage laws, *fails* in most contemporary high-speed circuits. These must be analyzed using electromagnetic field theory. Signal power flows are *not* confined to the intended metal wires or circuit paths.

On the other side of the electromagnetic spectrum, photonic devices, key components in current telecommunication networks, have to be analyzed more and more by means of full wave simulators. This is due to the dimensions of these devices which are no longer large in comparison to the used wavelength. All these changes result in the increasing need of efficient electromagnetic field solvers.

2 NUMERICAL TECHNIQUES

Nowadays, since most electromagnetic problems cannot be solved analytically, numerical simulation software is the only way to get to a solution. Several ways to solve Maxwell's equations numerically have been proposed,

| | MoM | FDTD | FEM |
|-------------------|----------|----------|----------|
| Grid uniformity | no | yes | no |
| Memory | small | huge | large |
| Open field | great | terrible | terrible |
| Sources | current | E, H | E, H |
| Dielectrics | terrible | great | great |
| freq./time domain | both | time | both |
| Non-linearity | terrible | great | terrible |

Table 1.1: A number of properties compared for different numerical techniques.

where each technique has its own merits and reason for existence, the best known and most popular are [2], [3]:

- BIE/MoM: Boundary Integral Equations and more specifically the solution with the Method of Moments (MoM) [4]. The MoM is very well suited for the analysis of antenna and scattering problems and passive microwave circuits. The MoM is not well suited for the analysis of complex geometries or inhomogeneous dielectrics.
- FEM: Finite Element Method [5]. The FEM is well suited for geometrically more complex structures. The mesh that is used does not need to be regular and the electric properties of each element can be chosen independently. On the other hand FEM has difficulties with open configurations.
- FDTD: Finite-Difference Time-Domain method [6]. The FDTD method is versatile and well suited for broadband problems. Furthermore it easily incorporates nonlinear materials or active elements. Disadvantages of the FDTD method are the inherent uniform grid, making simulations with curved boundaries and small features difficult.

In Table 1.1, some aspects of these different techniques are compared [7].

In the MoM technique the number of variables has to be limited since a linear system of equations, with full system matrices, has to be solved. Similarly for FEM, a linear system of equations has to be solved. However the system matrices are now sparse, thus allowing a higher number of variables. In the FDTD technique, the solution does not require the solution of a system of equations. The solution is calculated in an iterative fashion, where only neighbouring values, in space and time, are required. This allows the simulation of problems with a higher number of variables

as compared to the other methods. However, due to the uniform mesh, this higher number of field variables is also needed. A computational advantage of the FDTD technique is that it can readily be implemented on parallel computers. More details on these methods can be found in [8] and the references therein.

3 THE FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

In this thesis, only the Finite-Difference Time-Domain (FDTD) method will be studied. The FDTD method is a very active area of electromagnetic research and, compared to FEM and MoM, a younger area of research. This can be illustrated by the statistics maintained at www.fdt.org, a website entirely devoted to the FDTD method and containing primarily references to FDTD related publications in conference proceedings and in international papers. At this moment, October 2002, the database contains more than 2500 references to journal papers, about 2000 references to proceedings papers and 162 Ph. D. theses. The vast majority of these papers was published during the last ten years. At the Department of Information Technology at Ghent University, four Ph. D. theses were to a large extent devoted to the FDTD technique [9–12].

Although the FDTD method was presented for the first time in 1966 [13], it took several years before further research was started and only in the nineties the FDTD method has become an important area of research. The cause for this delay is highly linked with the advances of computer capabilities. Only about 15 years from now simple desktop PC's have sufficient memory to be able to use the FDTD method effectively. This consequently boosted the research for the FDTD method. With the development of techniques that helped to resolve some of the shortcomings of the FDTD method, the number of potential applications that could be analyzed with the method, and together with it the interest of scientists from outside the electromagnetics community, rose dramatically. The popularity of the method is also enhanced by the simplicity of the basic algorithm. All this resulted in the increasing interest in the FDTD method as a powerful full-wave technique for solving Maxwell's equations.

The popularity of the FDTD technique can also be measured by the number of commercially available FDTD simulation programs. Table 1.2 shows the programs that are based, either partially or entirely, on the FDTD technique. Along with the name of the program, the name of the company is

shown and a typical area of research where the simulation software can be used. These typical examples indicate what kind of purposes the software was designed for. Although the FDTD technique is highly application independent, the advanced techniques developed over the years to improve the FDTD technique are not. Often they are only of interest to a specific application: subcell models, e.g. for a thin wire, are of interest for electromagnetic interference (EMI) simulations whereas in the simulation of photonic devices the inclusion of nonlinear materials can be of major interest. Some software packages do not only offer advanced FDTD techniques, but offer important simulation aids, e.g. the software program SEMCAD can be purchased together with a number of human and animal phantoms, which can be used to investigate the biological effects of electromagnetic radiation.

In order to show the versatility of the method, some state-of-the-art applications, where the FDTD simulation technique plays a vital role, are given here:

- The specific absorption rate (SAR) distribution, giving the amount of energy absorbed in biological tissues and more specifically the human brain, when a mobile phone is held against the head, is mainly simulated with the FDTD method. A recent study showed good agreement between empirical and numerical results [14].
- Photonic crystal structures, which manipulate photons in a similar way as a semiconductor affects electrons, are newly developed materials. These photonic crystal structures give the ability to mold the light with new possibilities: sharp bends, channel-drop filters, (de)multiplexers, etc. [15]. An important analysis technique for the simulation of these structures is the FDTD method.
- The full wave analysis of planar interconnect structures performed using a SPICE circuit simulator in combination with the FDTD full wave simulation technique. The technique allows to simulate the crosstalk between two parallel lines. A typical study showed very good agreement with measured results [16].
- Breast cancer research. New microwave methods to detect malignant breast tumor are being developed. One possibility is called Confo-cal Microwave Imaging (CMI), where, using a number of microwave antenna and receivers, areas of significant scattering are localized. Since the electrical properties of healthy and malignant breast tissue contrast significantly, the areas where scattering is significant can be an indication of a breast tumor. The FDTD method was used in the testing of the feasibility of this approach [17].

| Name | Company | Typical application |
|-------------|--------------------------------|----------------------------------------|
| APLAC | Aplac Solution Corporation | circuit design |
| APSS | Apollo Photonics | photonic IC's |
| ApsimFDTD | Applied Simulation Technology | signal integrity EMI analysis |
| CFD-Maxwell | CFD Research Corporation | general problems |
| LC | Cray Research | electrical interconnects |
| MAFIA | Computer Simulation Technology | antennas bio-electromagnetics |
| EMA3D | Electromagnetic Applications | electromagnetic pulse |
| EZ-FDTD | EMS+ | EMI/EMC |
| Empire | IMST | antennas |
| SEMCAD | Schmid & Partner Engineering | antenna design bio-electromagnetics |
| Fullwave | Optima Research | photonic devices |
| OptiFDTD | Optiwave Corporation | photonic devices |
| QuickWave | QWED | antennas microwave components |
| FullWave | RSOFT Design Group | photonic Devices |
| XFDTD | REMCOM | bio-electromagnetics microstrip |
| Concerto | Vectorfields | antennas |
| Celia | Virtual Science | ground penetrating radar |
| Fidelity | Zeland Software | microwave circuits / EMC |

Table 1.2: List of commercially available software programs based, either entirely or partly, on the FDTD method.

Since research in the FDTD method is still in full progress, it can be expected that the FDTD method will play an important role in electromagnetics in the future. In [18] the future of FDTD computational electromagnetics is outlined and this is based on two improvements: expected advances in computer capabilities and the current advances in FDTD theory and algorithms. Some of these emerging prospects are:

- simulation of electrically large structures, which could be used for the detection of buried structures, e.g. landmine detection, and wireless signal propagation within buildings
- simulation of structures having both coarse and fine features, which could be used to analyze integrated electronic circuits, several wave-

lengths in dimension but with critical structures of 0.01λ

- long time simulations, which could be used to simulate very low frequency bioelectric phenomena, e.g. a neuromuscular pulse
- inverse and imaging problems, which could be used for geophysical prospecting and medical imaging

4 OBJECTIVES

One important aspect of the standard FDTD method is the uniform grid on which it is based. In the examples mentioned in the previous section, the FDTD method is working very well, since in that case the uniform grid is not a drawback. When the effect of microwaves on the human body needs to be investigated, a numerical model of the human body is required. Since the FDTD technique works easily with a large number of variables, certainly a lot more than FEM or MoM, where each cell has its own characteristic electric properties, the uniformity of these cells does not pose a problem, as a more general grid would not easily lead to a better model. In the example of the photonic crystals, the uniform mesh is not a problem since the dielectric rods, placed on the crossing of a regular square lattice, have dimensions corresponding to the FDTD grid dimensions.

However, in a lot of problems, the uniform mesh is not an adequate solution. When a small geometric feature needs to be incorporated into a simulation working with a uniform grid, the dimensions of the uniform cells of the grid have to be reduced to the dimension of the smallest geometric feature. Due the uniformity of the grid, these small cells have to be used throughout the entire simulation domain, including the region where a coarser grid would be sufficient, thus leading to excessive computational requirements.

Several solutions have been proposed to overcome this problem. These methods can be divided into two classes: subgridding and subcellular techniques. In the first class, the subgridding techniques, certain designated subdomains are gridded more finely, referred to as the fine grid, than the rest of the problem space, referred to as the coarse grid. The key aspect of these techniques is the coupling of the fine and the coarse grid. This coupling needs to be done both spatially and in the time domain. Spatially, at the boundary between the fine and the coarse grid, the higher number of fine grid field variables needs to be connected to the lower number of coarse grid field variables. The spatial coupling is mostly based on interpolation. The coupling in time is also required, since the marching in time

on the fine grid and the coarse grid does not correspond. This coupling in time often requires some kind of extrapolation or use of some kind of artificial wave equation. The advantage of subgridding techniques is the generality of the approach: no assumptions are made about the contents of the fine grid. A drawback is the complexity of most techniques and the often occurring late-time instability. In the first chapter of [19], by K. L. Shlager and J. B. Schneider, a survey of the finite-difference time-domain literature is given. In it, a section is devoted to subgridding techniques up to 1998, and several references regarding the subject have been added. In Chapter 2 of this Ph. D. thesis an overview of the existing subgridding techniques will be given.

The other class of techniques to incorporate small features in the simulation domain are the subcellular techniques. In these approaches, based on known analytical or static behaviour of small geometrical features, the equations for the field variables surrounding these fine grained structural features are modified. The advantage of these techniques is the simplicity of the approach. Only a limited number of update equations has to be modified and the rest of the FDTD algorithm does not change. Marching in time goes very naturally. The drawbacks are the limited number of subcell models that exist: thin wires, thin slots, narrow apertures, etc., and also sometimes the late-time instability. In the first chapter of [19], the subcellular techniques are listed and several references are given. In Chapter 6 of this Ph. D. thesis a short overview of the existing subcellular techniques is provided.

The need for FDTD techniques that combine the advantages of both approaches is obvious. A technique that on the one hand combines the generality of subgridding approaches and on the other hand the simplicity of the subcellular algorithms is highly desirable. The late-time stability is another important aspect, which needs to be improved or controlled if possible. The algorithms presented in this work have been developed with this in mind. It has led to the introduction of a new algorithm called the *subdomain FDTD method* which can be considered as a generalized subcellular technique, but based on subgridding techniques.

5 OUTLINE OF THIS WORK

In this work two novel FDTD algorithms will be proposed: the subdomain FDTD method and the generalized FDTD method. The subdomain FDTD method is of importance because it is a more general subcellular technique, compared to older subcellular techniques. One aspect of the standard

FDTD method is its simplicity, in other words it does not involve linear algebra. In this work however linear algebra will play a major role. On the one hand the proposed algorithms use some general algebraic routines: reduced order modeling (ROM) and subdomain time discretization. On the other hand, equally important, the use of linear algebra allows us to understand some important properties of the FDTD algorithms and clarifies the relation between different kinds of time discretization. In particular it helps to understand stability and late-time instability.

In Chapter 2, starting from Maxwell's equations in two dimensions (2D), spatial discretization, as used in the FDTD method, is discussed. The analysis will not be restricted to a uniform grid, but will focus on subgridding, which is a combination of a uniform fine grid and a uniform coarse grid. The uniform grid, as used in the standard FDTD method, is second order accurate. This means that each approximation, which typically results from a central difference approximation of the derivatives is second order accurate. It will be shown that for subgridding this second order accuracy can also be maintained. The concept of spatial reciprocity, an important property with respect to the stability of FDTD methods, is put forward. In the remainder of the chapter the state-space model and subdomain concepts are introduced. Such a subdomain is a part of the entire grid. Two specific subdomain types will be introduced: the E -type subdomain with only electric fields at the boundary, and the H -type subdomain with only magnetic fields at the boundary.

In Chapter 3 the next discretization step is discussed: temporal discretization. Different ways to discretize time are considered. A first implementation leads to the standard FDTD method. This is a leapfrog algorithm. A second implementation leads to a method called alternating direction implicit FDTD (ADI-FDTD) method. The ADI-FDTD method is not a leapfrog method. These techniques can only work with a uniform grid. In addition to that, a more general leapfrog time discretization method is introduced. This technique does not only work with field variables, but combines both field variables and more general subdomains. This method has been called the subdomain FDTD method. Its main characteristic is the alternating way, hence leapfrog, to first update the electric fields and the E -type subdomains and then the magnetic fields and the H -type subdomains. Inside the subdomains the grid can be based on subgridding. When only subdomains are present, the method is called the *generalized subdomain FDTD method*. If the subdomains have some characteristics just as the black and white squares of a chessboard have, then timestepping works by loop updating first the white areas and then the black areas. The term generalized refers to the fact that the fields at the subdomain boundary no

longer have to be solely electric or magnetic. All these time discretization methods are second order accurate.

To improve the efficiency of the new algorithms an important step has to be added before the subdomain models are time discretized: a reduced order model (ROM) of each subdomain has to be generated. A ROM algorithm replaces a state-space model with a large number of internal variables, by an approximate model with a smaller number of new internal variables. In Chapter 4 some ROM algorithms are discussed. Furthermore, a survey of the literature combining the FDTD method with ROM algorithms is given. The chapter is concluded with a summary of the steps of the new subdomain FDTD algorithms.

In Chapter 5 we discuss an important aspect related to explicit algorithms: stability. Three possible causes of instability can be distinguished. These three causes correspond with the three major topics discussed in the previous chapters. First of all, in the spatial discretization step, it will be shown that when spatial reciprocity is lost, it is no longer possible to guarantee any form of stability. Secondly, during the ROM step, the appropriate form of the state-space model needs to be used, otherwise the resulting model is highly unstable. Thirdly, as a consequence of time discretization, a maximum value for the time step in the standard FDTD method will be derived. This time step is better known as the Courant limit, and is a result of the explicit method used for the time discretization. For the less known ADI-FDTD method this condition does not apply, therefore this method is unconditionally stable. Finally, for the novel subdomain FDTD method, a condition related to the stability of a standard FDTD problem is derived. When during one of these three steps stability is lost, the subsequent steps cannot remedy for it.

Finally, in Chapter 6, numerical results for the newly developed simulation technique are evaluated. First the generalized subdomain FDTD method is validated. Afterwards the subdomain FDTD method is investigated: some examples of general subcell models are analyzed. The first example studies the effect of using the combination of a fine grid and a coarse grid. Further on thin wire models of a perfectly conducting wire and a dielectric wire are investigated. Other analyzed models have been chosen to show the versatility of the approach: these numerical examples include asymmetrical features, lossy materials and geometries where the material boundary intersects the fine grid – coarse grid boundary. The chapter is closed with the analysis of a photonic crystal waveguide, where the same model is used about 2000 times in one simulation. For several examples the efficiency of the new approach, compared to the standard FDTD method, is given.

The most important new contributions that will be presented in this work are:

1. The introduction of the subdomain FDTD method, which can be considered as a technique to generate subcell models in an automatic and general way. The approach makes use of an advanced new ROM algorithm [20] and it will be shown that this new approach is versatile and very efficient.
2. The introduction of the generalized subdomain FDTD method, which creates macromodels by means of ROM based on an FDTD grid and uses a time iteration scheme that has the same characteristics as time stepping in the FDTD method.
3. A rigorous study of the stability of these new methods and some implications for the standard FDTD method and for the recent ADI-FDTD method.

To conclude this introductory chapter an overview of the publications, which resulted from this research, are given. The research resulted in two publications, both as first author, in international journals [21, 22], five articles in the proceedings of international conferences [23–28] and one abstract of an international conference [29]. Finally there was one article in the proceedings of a national conference [30].

BIBLIOGRAPHY

- [1] A. Taflove, “Why study electromagnetics: The first unit in an undergraduate electromagnetics course,” *IEEE Antennas Propagat. Magazine*, vol. 44, no. 2, pp. 132–139, Apr. 2002.
- [2] R. Marg, G. Oberschmidt, and A. F. Jacob, “Numerical techniques for microwave circuits,” *IEEE - MTT/AP German Newsletter*, vol. 2, no. 2, Oct. 1998.
- [3] T. H. Hubing, “Survey of numerical electromagnetic modeling techniques,” *Technical Report TR91-1-001.3, Dept. of Electrical Engineering, University of Missouri-Rolla*, Sep. 1991.
- [4] R. F. Harrington, *Field Computations by Moment Methods*, Macmillan, New York, 1968.
- [5] J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 1993.

-
- [6] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 1995.
 - [7] L. Daniel, "Simulation for signal integrity and electromagnetic interference," *Taken from course EE219A at University of California, Berkeley*.
 - [8] H. Rogier, *Numerieke oplossing van de Maxwell vergelijkingen door combinatie van eindige differenties of eindige elementen met integraalvergelijkingen*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1999.
 - [9] M. De Pourcq, *Theoretische en experimentele bijdragen tot de mikrogolfverwarming*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1985.
 - [10] J. Van Hese, *Analyse- en designmethodes voor impedantiegecontroleerde overgangsstructuren in hoge snelheid transmissiesystemen*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1992.
 - [11] J. De Moerloose, *FDTD-technieken voor open gebieden met toepassingen op antennemodellering*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1994.
 - [12] S. Van den Berghe, *Object-oriented Electromagnetic Simulations with the Finite-Difference Time-Domain Method*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1999.
 - [13] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302–307, May 1966.
 - [14] P. Gajšek, T. J. Walters, W. D. Hurt, J. M. Ziriak, D. A. Nelson, and P. A. Mason, "Empirical validation of SAR values predicted by FDTD modeling," *Bioelectromagnetics*, vol. 23, no. 1, pp. 37–48, January 2002.
 - [15] S. G. Johnson, A. Mekis, S. Fan, and J. D. Joannopoulos, "Molding the flow of light," *Computing in Science & Engineering*, vol. 3, no. 6, pp. 38–47, Nov. 2001.

- [16] N. Orhanovic, R. Raghuram, and N. Matsui, "Full wave analysis of planar interconnect structures using FDTD-SPICE," *IEEE Electronic Components and Technology Conference*, pp. 489-494, May 2001.
- [17] E. C. Fear, S. C. Hagness, P. M. Meaney, M. Okoniewski, and M. A. Stuchly, "Enhancing breast tumor detection with near-field imaging," *IEEE Microwave Magazine*, vol. 3, no. 1, pp. 48-56, Mar. 2002.
- [18] A. Taflove, "Emerging prospects for FDTD computational electromagnetics," *Int. Conference on Electromagnetics in Advanced Applications*, pp. 345-348, Sep. 2001.
- [19] A. Taflove, *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 1998.
- [20] L. Knockaert and D. De Zutter, "Laguerre-SVD reduced-order modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1469-1475, Sep. 2000.
- [21] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," *IEEE Microwave and Guided Wave Letters*, vol. 10, no. 8, pp. 301-303, Aug. 2000.
- [22] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Generation of FDTD subcell equations by means of reduced order modeling," *IEEE Trans. Antennas Propagat.*, accepted for publication.
- [23] B. Denecker, D. De Zutter, L. Knockaert, and F. Olyslager, "A higher level algorithm for 2D electromagnetic modelling using an FDTD grid," *IEEE Antennas Propagat. Symp.*, vol. 3, pp. 1340-1343, July 2000.
- [24] B. Denecker, F. Olyslager, D. De Zutter, and L. Knockaert, "2-D FDTD subgridding based on subdomain generation," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 288-290, May 2001.
- [25] B. Denecker, F. Olyslager, D. De Zutter, L. Klinkenbusch, and L. Knockaert, "Efficient analysis of photonic crystal structures using a novel FDTD-technique," *IEEE Antennas Propagat. Symp.*, vol. 4, pp. 344-347, June 2002.
- [26] L. Knockaert and B. Denecker, "Explicit reciprocity and reduced order modeling," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 497-499, May 2001.

- [27] L. Knockaert, B. Denecker, and D. De Zutter, "Explicitly reciprocal reduced order modeling: Laguerre-SVD versus balanced realizations," *IEEE Antennas Propagat. Symp.*, vol. 2, pp. 556–558, June 2002.
- [28] L. Knockaert, B. Denecker, D. De Zutter, and F. Olyslager, "Reduced order multiport modelling via Laguerre-SVD and its application to FDTD," *XXVIIth URSI General Assembly, Maastricht, The Netherlands*, pp. CD-ROM, Aug. 2002.
- [29] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Efficient FDTD analysis of very large finite photonic crystal structures," *LEOS Benelux, Photonic Crystal Workshop, Ghent*, May 2002.
- [30] B. Denecker, "The finite-difference time-domain method and reduced order modelling: Electromagnetics," *1e Doctoraatssymposium FTW, Gent*, p. 9, Dec. 2000.

CHAPTER 2

Finite differences: the spatial problem

1 INTRODUCTION

Over the years the finite-difference time-domain (FDTD) method has become an important and popular simulation technique for electromagnetics [1]. Its importance is illustrated by the existence of a web-site, www.fdt.org, which is a large searchable database only containing references to papers related to the FDTD method. The popularity is mainly brought about thanks to the fact that the method is conceptually easy. Also, as far as programming is concerned, the FDTD method is, at least in the beginning, straightforward and requires limited effort to get a first, albeit small, program running.

Despite all this, the FDTD method also has some major drawbacks: the grid as presented initially by Yee in 1966 [2] is based on a regular orthogonal grid, where the material properties are supposed to be constant in each cell. In this way small geometric features can only be simulated by using a very small grid, even if the physics of the problem only requires this in a small part of the problem space. Another drawback is the requirement of an orthogonal grid, which implies that round or skew boundaries are approximated by a piecewise constant curve.

In this chapter basic aspects of the FDTD method will be explained, where we will restrict ourselves to two dimensions (2D). In this way, all the important concepts inherent to the FDTD method and to the algorithms will become clear avoiding unnecessary complexity. As we go along, matrix notation will prove to be of major importance.

The question that will be answered in this chapter is the following: “How

can a real world problem, continuous in both space and time, as a first step be transformed into an analogous problem in a discrete world, where time is still continuous?" This means that space will be discretized and throughout this chapter time will remain a continuous variable. In the following chapter, the step from a continuous time problem to a discrete time problem will be explained. The discretized space will first of all be the regular orthogonal grid, as it was introduced by Yee (Section 2). In a following section subgridding will be treated. This is a mixture of a coarse grid and a local fine grid. In a final section, the resulting equations will be written in a general matrix form: the state-space model.

2 UNIFORM ORTHOGONAL GRID

2.1 Introduction

As a starting point, for discretizing the spatial derivatives of Maxwell's equations, the seminal work for the FDTD method by Yee [2] is given. The main idea is to approximate the derivatives by second order central differences. By properly choosing the locations where this is done, namely at uniformly spaced points, the resulting set of *field variables* becomes a good approximation of the continuous fields. A field variable is a field component in such a point. The different field components of the electric and magnetic field, are not chosen in the same point but are located at interleaved points. This results in higher accuracy with the same number of field variables.

Before the spatial derivatives of the 2D Maxwell's equations are discretized, the TE-case and the TM-case are written down in a more general form in subsection 2.2. Then in subsection 2.3 some notational grid related aspects are introduced, which will be followed in subsection 2.4 by the discretization of the spatial derivatives. Finally in subsection 2.5 some aspects concerning the nature of the obtained equations are discussed.

2.2 Generalized 2D Maxwell's equations

The basis of electromagnetism are Maxwell's equations. In differential form, the most interesting form to explain the discretization, they are given

by:

$$\nabla \times \mathbf{E}(\mathbf{r}, t) = -\frac{\partial \mathbf{B}(\mathbf{r}, t)}{\partial t} - \mathbf{J}_m(\mathbf{r}, t) \quad (2.1a)$$

$$\nabla \times \mathbf{H}(\mathbf{r}, t) = \frac{\partial \mathbf{D}(\mathbf{r}, t)}{\partial t} + \mathbf{J}_e(\mathbf{r}, t) \quad (2.1b)$$

$$\nabla \cdot \mathbf{B}(\mathbf{r}, t) = \rho_m(\mathbf{r}, t) \quad (2.1c)$$

$$\nabla \cdot \mathbf{D}(\mathbf{r}, t) = \rho_e(\mathbf{r}, t) \quad (2.1d)$$

Here, \mathbf{E} is the *electric field* vector [V/m], \mathbf{H} the *magnetic field* vector [A/m], \mathbf{D} the *electric flux density* vector [C/m²], \mathbf{B} the *magnetic flux density* vector [Wb/m²], \mathbf{J}_e the *electric conduction current density* vector [A/m²], \mathbf{J}_m the *magnetic conduction current density* vector [V/m²], ρ_e the *electric charge density* [C/m³] and ρ_m the *magnetic charge density* [Wb/m³]. No electric or magnetic current sources were considered. The use of sources will not be studied here, for a thorough explanation we refer to [3].

We will limit ourselves to the use of linear, isotropic and nondispersive materials. For these materials, \mathbf{B} and \mathbf{H} are related by:

$$\mathbf{B}(\mathbf{r}, t) = \mu(\mathbf{r})\mathbf{H}(\mathbf{r}, t) \quad (2.2a)$$

and \mathbf{D} and \mathbf{E} are related by:

$$\mathbf{D}(\mathbf{r}, t) = \epsilon(\mathbf{r})\mathbf{E}(\mathbf{r}, t) \quad (2.2b)$$

where $\epsilon = \epsilon_0\epsilon_r$ is the *electric permittivity* or *dielectric constant* [F/m] and $\mu = \mu_0\mu_r$ is the *permeability* [H/m]. Often the dimensionless quantities ϵ_r , the *relative dielectric constant*, and μ_r , the *relative permeability* will be used. They are proportional to the free space dielectric constant and permeability:

$$\epsilon_0 = 8.854 \times 10^{-12} \text{ F/m} \quad (2.3a)$$

$$\mu_0 = 4\pi \times 10^{-7} \text{ H/m} \quad (2.3b)$$

In a lossy medium the electric current density \mathbf{J}_e accounts for the electric losses:

$$\mathbf{J}_e(\mathbf{r}, t) = \sigma_e(\mathbf{r})\mathbf{E}(\mathbf{r}, t) \quad (2.4a)$$

where σ_e is the *electric conductivity* [S/m] of the medium. Permitting the possibility of magnetic losses, the magnetic current density \mathbf{J}_m can be defined analogously:

$$\mathbf{J}_m(\mathbf{r}, t) = \sigma_m(\mathbf{r})\mathbf{H}(\mathbf{r}, t) \quad (2.4b)$$

where σ_m is the *magnetic conductivity* [Ω /m].

In this work, as a further simplification, only the two dimensional (2-D) case will be studied. For this we assume that the problem space is invariant in the z -direction: neither the excitation nor the modeled geometry has any variation in the z -direction. Applying this, namely each z -derivative is identically zero, (2.2) and (2.4) to the first two equations of the 3-D Maxwell's equations, (2.1a) and (2.1b), it is possible to split the resulting equations into two groups. The first group only involves the components E_x , E_y and H_z :

$$\epsilon(\mathbf{r}) \frac{\partial E_x(\mathbf{r}, t)}{\partial t} = \frac{\partial H_z(\mathbf{r}, t)}{\partial y} - \sigma_e(\mathbf{r}) E_x(\mathbf{r}, t) \quad (2.5a)$$

$$\epsilon(\mathbf{r}) \frac{\partial E_y(\mathbf{r}, t)}{\partial t} = -\frac{\partial H_z(\mathbf{r}, t)}{\partial x} - \sigma_e(\mathbf{r}) E_y(\mathbf{r}, t) \quad (2.5b)$$

$$\mu(\mathbf{r}) \frac{\partial H_z(\mathbf{r}, t)}{\partial t} = \frac{\partial E_x(\mathbf{r}, t)}{\partial y} - \frac{\partial E_y(\mathbf{r}, t)}{\partial x} - \sigma_m(\mathbf{r}) H_z(\mathbf{r}, t) \quad (2.5c)$$

and is called the *transverse electric* (TE) case. The second group only contains H_x , H_y and E_z :

$$\mu(\mathbf{r}) \frac{\partial H_x(\mathbf{r}, t)}{\partial t} = -\frac{\partial E_z(\mathbf{r}, t)}{\partial y} - \sigma_m(\mathbf{r}) H_x(\mathbf{r}, t) \quad (2.6a)$$

$$\mu(\mathbf{r}) \frac{\partial H_y(\mathbf{r}, t)}{\partial t} = \frac{\partial E_z(\mathbf{r}, t)}{\partial x} - \sigma_m(\mathbf{r}) H_y(\mathbf{r}, t) \quad (2.6b)$$

$$\epsilon(\mathbf{r}) \frac{\partial E_z(\mathbf{r}, t)}{\partial t} = -\frac{\partial H_x(\mathbf{r}, t)}{\partial y} + \frac{\partial H_y(\mathbf{r}, t)}{\partial x} - \sigma_e(\mathbf{r}) E_z(\mathbf{r}, t) \quad (2.6c)$$

and is called the *transverse magnetic* (TM) case.

These two cases are mathematically very similar, consider for example (2.5a) and (2.6a):

$$\epsilon(\mathbf{r}) \frac{\partial E_x(\mathbf{r}, t)}{\partial t} = \frac{\partial H_z(\mathbf{r}, t)}{\partial y} - \sigma_e(\mathbf{r}) E_x(\mathbf{r}, t) \quad (2.7)$$

$$\mu(\mathbf{r}) \frac{\partial H_x(\mathbf{r}, t)}{\partial t} = -\frac{\partial E_z(\mathbf{r}, t)}{\partial y} - \sigma_m(\mathbf{r}) H_x(\mathbf{r}, t) \quad (2.8)$$

Both equations are of the following form:

- The left hand side contains the time derivative of the x -directed field component multiplied by some material specific parameter.
- The first term of the right hand side is the y -derivative of the z -directed field component.
- The second term of the right hand side is the x -directed field component multiplied by some loss factor.

Similar observations can be made for (2.5b & 2.6b) and for (2.5c & 2.6c). Thanks to these similarities both sets of equations can be expressed as two variations of one general form. Prior to introducing this general form, the equations are normalized. The TE-equations can be written in the following form:

$$\epsilon_r(\mathbf{r}) \frac{\partial (\sqrt{\epsilon_0} E_x(\mathbf{r}, t))}{\partial t} = c_0 \frac{\partial (\sqrt{\mu_0} H_z(\mathbf{r}, t))}{\partial y} - \frac{\sigma_e(\mathbf{r})}{\epsilon_0} (\sqrt{\epsilon_0} E_x(\mathbf{r}, t)) \quad (2.9a)$$

$$\epsilon_r(\mathbf{r}) \frac{\partial (\sqrt{\epsilon_0} E_y(\mathbf{r}, t))}{\partial t} = -c_0 \frac{\partial (\sqrt{\mu_0} H_z(\mathbf{r}, t))}{\partial x} - \frac{\sigma_e(\mathbf{r})}{\epsilon_0} (\sqrt{\epsilon_0} E_y(\mathbf{r}, t)) \quad (2.9b)$$

$$\begin{aligned} \mu_r(\mathbf{r}) \frac{\partial (\sqrt{\mu_0} H_z(\mathbf{r}, t))}{\partial t} = c_0 \left[\frac{\partial (\sqrt{\epsilon_0} E_x(\mathbf{r}, t))}{\partial y} - \frac{\partial (\sqrt{\epsilon_0} E_y(\mathbf{r}, t))}{\partial x} \right] \\ - \frac{\sigma_m(\mathbf{r})}{\mu_0} (\sqrt{\mu_0} H_z(\mathbf{r}, t)) \end{aligned} \quad (2.9c)$$

and the TM-equations are:

$$\mu_r(\mathbf{r}) \frac{\partial (\sqrt{\mu_0} H_x(\mathbf{r}, t))}{\partial t} = -c_0 \frac{\partial (\sqrt{\epsilon_0} E_z(\mathbf{r}, t))}{\partial y} - \frac{\sigma_m(\mathbf{r})}{\mu_0} (\sqrt{\mu_0} H_x(\mathbf{r}, t)) \quad (2.10a)$$

$$\mu_r(\mathbf{r}) \frac{\partial (\sqrt{\mu_0} H_y(\mathbf{r}, t))}{\partial t} = c_0 \frac{\partial (\sqrt{\epsilon_0} E_z(\mathbf{r}, t))}{\partial x} - \frac{\sigma_m(\mathbf{r})}{\mu_0} (\sqrt{\mu_0} H_y(\mathbf{r}, t)) \quad (2.10b)$$

$$\begin{aligned} \epsilon_r(\mathbf{r}) \frac{\partial (\sqrt{\epsilon_0} E_z(\mathbf{r}, t))}{\partial t} = -c_0 \left[\frac{\partial (\sqrt{\mu_0} H_x(\mathbf{r}, t))}{\partial y} - \frac{\partial (\sqrt{\mu_0} H_y(\mathbf{r}, t))}{\partial x} \right] \\ - \frac{\sigma_e(\mathbf{r})}{\epsilon_0} (\sqrt{\epsilon_0} E_z(\mathbf{r}, t)) \end{aligned} \quad (2.10c)$$

Then, all these equations can be written in the following generalized form:

$$\theta_r(\mathbf{r}) \frac{\partial O_x(\mathbf{r}, t)}{\partial t} = -c_0 \frac{\partial P_z(\mathbf{r}, t)}{\partial y} - \sigma(\mathbf{r}) O_x(\mathbf{r}, t) \quad (2.11a)$$

$$\theta_r(\mathbf{r}) \frac{\partial O_y(\mathbf{r}, t)}{\partial t} = c_0 \frac{\partial P_z(\mathbf{r}, t)}{\partial x} - \sigma(\mathbf{r}) O_y(\mathbf{r}, t) \quad (2.11b)$$

$$\kappa_r(\mathbf{r}) \frac{\partial P_z(\mathbf{r}, t)}{\partial t} = c_0 \left[\frac{\partial O_y(\mathbf{r}, t)}{\partial x} - \frac{\partial O_x(\mathbf{r}, t)}{\partial y} \right] - \sigma^*(\mathbf{r}) P_z(\mathbf{r}, t) \quad (2.11c)$$

where for the TE-case:

$$\theta_r(\mathbf{r}) = \epsilon_r(\mathbf{r}) \quad (2.12a)$$

$$\kappa_r(\mathbf{r}) = \mu_r(\mathbf{r}) \quad (2.12b)$$

$$O_x(\mathbf{r}, t) = \sqrt{\epsilon_0} E_x(\mathbf{r}, t) \quad (2.12c)$$

$$O_y(\mathbf{r}, t) = \sqrt{\epsilon_0} E_y(\mathbf{r}, t) \quad (2.12d)$$

$$P_z(\mathbf{r}, t) = -\sqrt{\mu_0} H_z(\mathbf{r}, t) \quad (2.12e)$$

$$\sigma(\mathbf{r}) = \sigma_e(\mathbf{r}) / \epsilon_0 \quad (2.12f)$$

$$\sigma^*(\mathbf{r}) = \sigma_m(\mathbf{r}) / \mu_0 \quad (2.12g)$$

and for the TM-case:

$$\theta_r(\mathbf{r}) = \mu_r(\mathbf{r}) \quad (2.13a)$$

$$\kappa_r(\mathbf{r}) = \epsilon_r(\mathbf{r}) \quad (2.13b)$$

$$O_x(\mathbf{r}, t) = \sqrt{\mu_0} H_x(\mathbf{r}, t) \quad (2.13c)$$

$$O_y(\mathbf{r}, t) = \sqrt{\mu_0} H_y(\mathbf{r}, t) \quad (2.13d)$$

$$P_z(\mathbf{r}, t) = \sqrt{\epsilon_0} E_z(\mathbf{r}, t) \quad (2.13e)$$

$$\sigma(\mathbf{r}) = \sigma_m(\mathbf{r}) / \mu_0 \quad (2.13f)$$

$$\sigma^*(\mathbf{r}) = \sigma_e(\mathbf{r}) / \epsilon_0 \quad (2.13g)$$

From this point on, it is possible to continue with equations (2.11). At the appropriate point, the distinction between both cases will be made. For the sake of readability, the time dependency of the field components O_x, O_y and P_z is no longer explicitly noted in equation (2.11). This will be continued throughout the remainder of this chapter.

2.3 The grid

The original FDTD method is based on spatial discretization by a uniform orthogonal grid (see Fig. 2.1). In the term 'uniform orthogonal grid', the word *uniform* illustrates the periodic nature of the grid: an elementary cell can be defined (see Fig. 2.2), which represents the building block for the entire grid. The grid is a large collection of elementary cells repeated in both x - and y -direction. The word *orthogonal* refers to the perpendicular edges of this elementary cell.

Since this grid will be used very often, it is important to introduce an elegant notation, one that will emphasize the periodic nature of the grid. The size of the elementary cell is $\Delta_x \times \Delta_y$ (Fig 2.2) and cell (i, j) refers to the rectangle with following corner points:

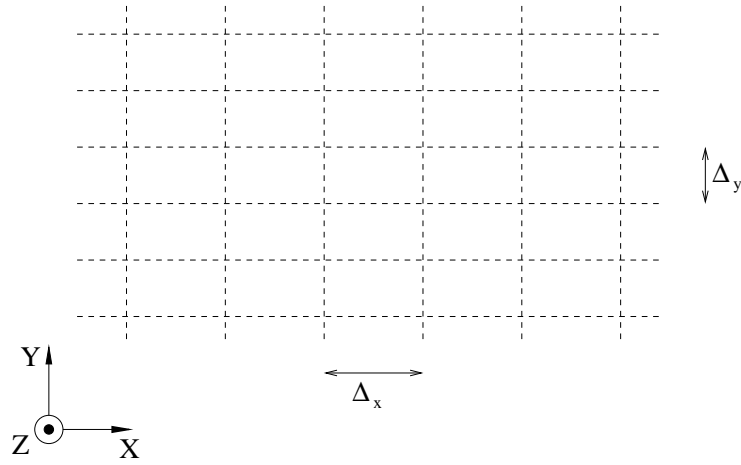


Figure 2.1: A uniform orthogonal grid.

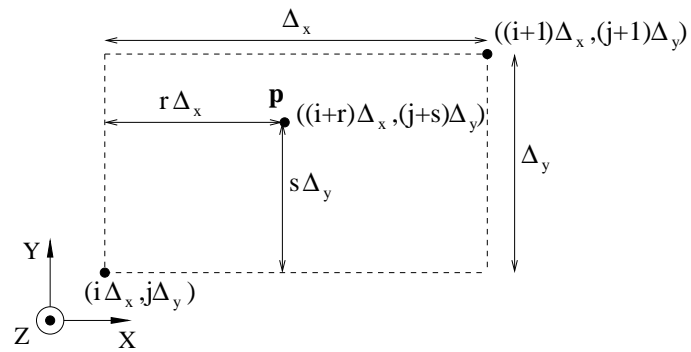


Figure 2.2: The elementary cell, the building block of the uniform orthogonal grid.

- left under corner point: $(i\Delta_x, j\Delta_y)$
- right upper corner point: $((i+1)\Delta_x, (j+1)\Delta_y)$

When a point \mathbf{p} is located inside the cell (i, j) (Fig. 2.2), then its location can be written as follows:

$$((i+r)\Delta_x, (j+s)\Delta_y) \quad \text{where } i, j \in \mathbb{N} \text{ and } 0 \leq r, s < 1 \quad (2.14)$$

A field component, say O_x , located at point \mathbf{p} will, from now on, be referred to as:

$$O_x|_{i+r, j+s} \quad (2.15)$$

this avoids repeating the constants Δ_x, Δ_y time and again. This kind of notation was already proposed by Yee [2].

2.4 Discretizing space

The original FDTD method is based on the use of a regular orthogonal grid. Therefore, the partial differential equations (2.11) will be approximated by their spatially discretized form at the same relative location inside each elementary cell. In other words at $(i+r, j+s)$, with r and s constant for each field component.

A difference approximation is merely the approximation of a function or the approximation of the derivative of a function by an algebraic sum of some function values. In the FDTD method the spatial derivatives are discretized using *central differences*. When two function values are at equal distance from the point where a derivative needs to be approximated, the scaled difference is a central difference approximation (Fig 2.3) of this derivative. For a function f this is:

$$\left. \frac{df(x)}{dx} \right|_{x=x_0} \approx \frac{f(x_0+d) - f(x_0-d)}{2d} \quad (2.16)$$

This case, where the difference at two equidistant points is taken, is very important since the approximation is second order accurate. This can easily be illustrated by considering Taylor's series expansion around $x = x_0$ for both values:

$$\begin{aligned} f(x_0-d) = f(x_0) - \frac{d}{1!}f'(x) \Big|_{x=x_0} + \frac{d^2}{2!}f''(x) \Big|_{x=x_0} \\ - \frac{d^3}{3!}f^{(3)}(x) \Big|_{x=x_0} + \dots \end{aligned} \quad (2.17a)$$

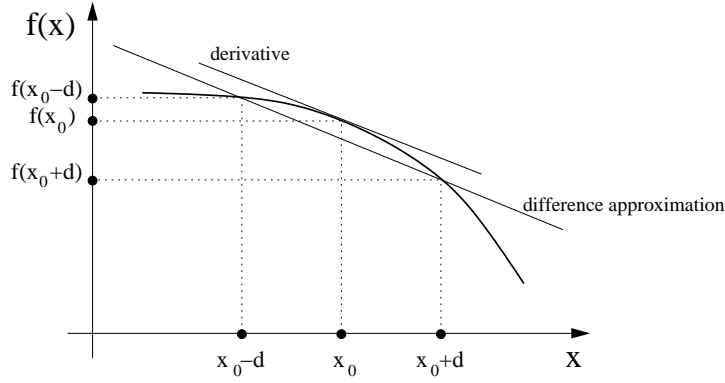


Figure 2.3: Graphical interpretation of a derivative and its central difference approximation.

$$f(x_0 + d) = f(x_0) + \frac{d}{1!} f'(x) \Big|_{x=x_0} + \frac{d^2}{2!} f''(x) \Big|_{x=x_0} + \frac{d^3}{3!} f^{(3)}(x) \Big|_{x=x_0} + \dots \quad (2.17b)$$

subtracting (2.17a) from (2.17b) and dividing by $2d$ then gives:

$$\frac{f(x_0 + d) - f(x_0 - d)}{2d} = f'(x) \Big|_{x=x_0} + \frac{d^2}{3!} f^{(3)}(x) \Big|_{x=x_0} + \dots \quad (2.18)$$

clearly showing the second order accuracy of the approximation.

Using central differences, the spatial derivatives in (2.11) can now be discretized. The following properties need to be maintained:

- The same field variables need to be used: Writing (2.11c) at $(i+r, j+s)$ uses $P_z(i+r, j+s)$ as a field variable. When central difference approximations are used, to approximate the spatial derivatives, with neighbours at distance d_x in the x -direction and d_y in the y -direction, then $O_x(i+r, j+s \pm d_y)$ and $O_y(i+r \pm d_x, j+s)$ are four field variables that also appear in the equation. Since we would like to use the same field variables, (2.11a) will be discretized at $(i+r, j+s \pm d_y)$, subsequently requiring a central difference approximation for $\partial P_z / \partial y$ with neighbours at distance d_y . In this manner $P_z(i+r, j+s)$ is employed again. Similarly for (2.11b) using d_x in the central difference approximation of $\partial P_z / \partial x$.
- Not all field components of an elementary cell, have to be discretized

in the same location. Accuracy prevails. This allows an interleaved position of O_x , O_y and P_z in each cell.

- The discretized problem space will be the regular grid (Fig. 2.1).

Let us start with (2.11c) and approximate the spatial derivatives in the point (x_0, y_0) using neighbouring points at a distance d_x in the x -direction and d_y in the y -direction:

$$\kappa_r|_{x_0, y_0} \frac{d P_z|_{x_0, y_0}}{dt} \approx c_0 \left[\frac{O_y|_{x_0+d_x, y_0} - O_y|_{x_0-d_x, y_0}}{2d_x} - \frac{O_x|_{x_0, y_0+d_y} - O_x|_{x_0, y_0-d_y}}{2d_y} \right] - \sigma^*|_{x_0, y_0} P_z|_{x_0, y_0} \quad (2.19)$$

Now, for the equation (2.11b), apply the central difference approximation in the point $(x_0 - d_x, y_0)$, since this results in the use of the same field variables ($O_y|_{x_0-d_x, y_0}$) in both equations. The distance at which the neighbours, used in the central difference approximation of the x -derivative, are evaluated will again be d_x :

$$\theta_r|_{x_0-d_x, y_0} \frac{O_y|_{x_0-d_x, y_0}}{\partial t} = c_0 \left[\frac{P_z|_{x_0, y_0} - P_z|_{x_0-2d_x, y_0}}{2d_x} \right] - \sigma|_{x_0-d_x, y_0} O_y|_{x_0-d_x, y_0} \quad (2.20)$$

since this results in the use of the same P_z field variable: $P_z|_{x_0, y_0}$. Applying the same reasoning for (2.11a) we obtain, by discretizing around the point $(x_0, y_0 - d_y)$:

$$\theta_r|_{x_0, y_0-d_y} \frac{O_x|_{x_0, y_0-d_y}}{\partial t} = -c_0 \left[\frac{P_z|_{x_0, y_0} - P_z|_{x_0, y_0-2d_y}}{2d_y} \right] - \sigma|_{x_0, y_0-d_y} O_x|_{x_0, y_0-d_y} \quad (2.21)$$

When the starting point is chosen to be the middle point of our elementary grid cell: $(i + 1/2, j + 1/2)$, then the location of the field variables in the grid is shown in Fig. 2.4.

When the same relative location inside each elementary cell is desired, then a proper choice for d_x and d_y has to be made. A first possibility would be $d_x = \Delta_x$. Then each elementary cell contains the 3 field components (O_x, O_y, P_z) at the same location: $(i + 1/2, j + 1/2)$. However, since the error of the central difference approximation is of the order of $\mathcal{O}(d^2)$, see (2.18), it is important to choose d_x and d_y as small as possible. Therefore an

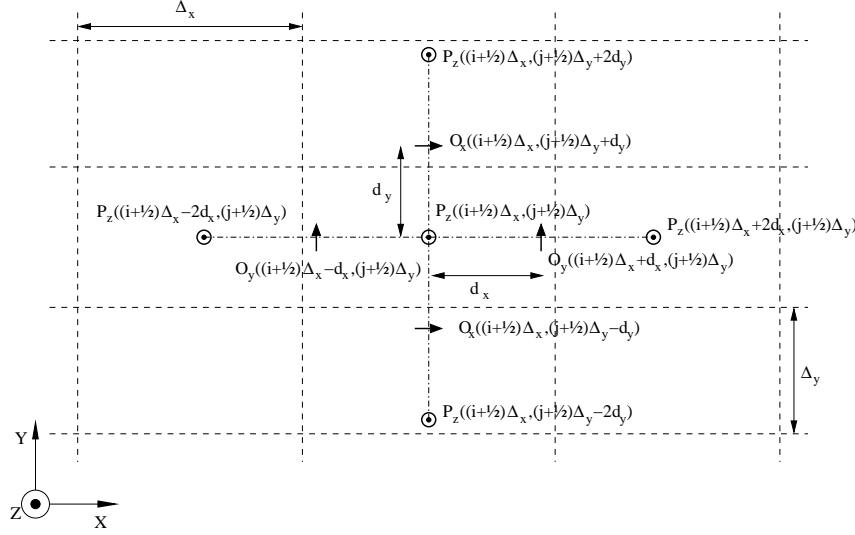


Figure 2.4: Location of field variables when discretized using central differences and distance d_x in x -direction and d_y in y -direction.

other choice is made: $d_y = \Delta_y/2$ and $d_x = \Delta_x/2$. Here again the 3 field components appear once per elementary cell, yet this time they are no longer located at the same position (see Fig. 2.5). This type of cell where the field components are interleaved in each elementary cell is called a Yee cell.

Although both choices have second order accurate approximations and 3 field variables per cell, the latter one is the best one since the error is four times as small. Choosing d_x and d_y even smaller is possible: $d_x = \Delta_x/n$ and $d_y = \Delta_y/n$. But this results only in a grid with smaller elementary cells, with again 3 field components per elementary cell. For n even, these field locations are located at the same position, for n odd they are located at interleaved points. The location of the different field components is now:

- $O_x|_{(i+1/2)\Delta_x, j\Delta_y}$ or in short hand notation: $O_x|_{i+1/2, j}$
- $O_y|_{i\Delta_x, (j+1/2)\Delta_y}$ or in short hand notation: $O_y|_{i, j+1/2}$
- $P_x|_{(i+1/2)\Delta_x, (j+1/2)\Delta_y}$ or in short hand notation: $P_x|_{i+1/2, j+1/2}$

The cell as introduced in Fig. 2.5 and the relative position of the field components inside this cell, is only one kind of representation. An alternative representation would be to start from an elementary cell corner point.

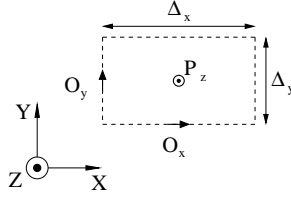


Figure 2.5: A 2D Yee grid cell indicating the location of the different field components inside each cell.

Shifting in this way the field components by $(-\frac{1}{2}\Delta_x, -\frac{1}{2}\Delta_y)$. This representation is not very important, as long as it is kept in mind that a grid does not have to be terminated by complete cells. The cells help to visualize the location of the field variables. The final equations are now:

$$\theta_r|_{i+1/2,j} \frac{d O_x|_{i+1/2,j}}{dt} = -c_0 \frac{P_z|_{i+1/2,j+1/2} - P_z|_{i+1/2,j-1/2}}{\Delta_y} - \sigma|_{i+1/2,j} O_x|_{i+1/2,j} \quad (2.22a)$$

$$\theta_r|_{i,j+1/2} \frac{d O_y|_{i,j+1/2}}{dt} = c_0 \frac{P_z|_{i+1/2,j+1/2} - P_z|_{i-1/2,j+1/2}}{\Delta_x} - \sigma|_{i,j+1/2} O_y|_{i,j+1/2} \quad (2.22b)$$

$$\kappa_r|_{i+1/2,j+1/2} \frac{d P_z|_{i+1/2,j+1/2}}{dt} = c_0 \left[\frac{O_y|_{i+1,j+1/2} - O_y|_{i,j+1/2}}{\Delta_x} - \frac{O_x|_{i+1/2,j+1} - O_x|_{i+1/2,j}}{\Delta_y} \right] - \sigma^*|_{i+1/2,j+1/2} P_z|_{i+1/2,j+1/2} \quad (2.22c)$$

The way the field components are spread over the regular grid (Fig. 2.1), is shown in Fig. 2.6

The Yee cell as presented in Fig. 2.5 is the 2D equivalent of the more general and more famous 3D Yee grid cell (Fig. 2.7), where each elementary cell has 6 field components.

2.5 Properties of spatially discretized equations

2.5.1 Characteristic structure

The equations as they are presented in (2.22) have some properties which are important and which will be used throughout the rest of the text. When

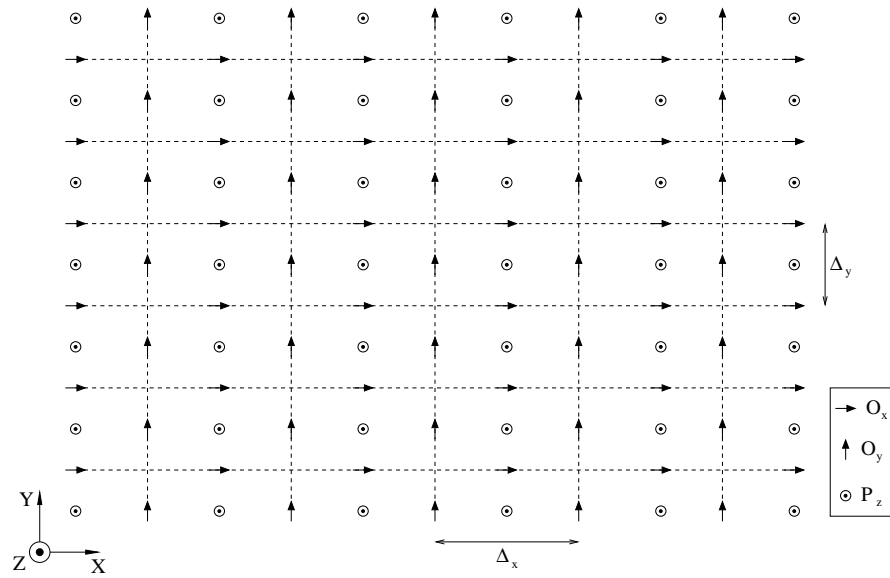


Figure 2.6: The regular uniform grid and the location of the different field components throughout the grid.

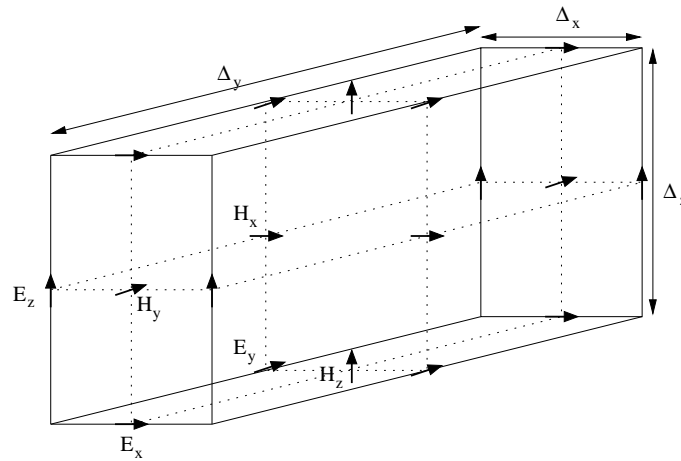


Figure 2.7: The 3D Yee cell indicating the position of the different field components.

looking at the equations (2.22), for each equation the following specific form can be observed:

- On the left side of the equation the time derivative of a field component (e.g. for equation (2.22a) $d O_x|_{i+1/2,j} / dt$) multiplied by a relative constitutive parameter ($\theta_r|_{i+1/2,j}$) is present. On the right hand side the same field component ($O_x|_{i+1/2,j}$) multiplied by a loss factor ($\sigma|_{i+1/2,j}$) is seen.
- In addition the right hand side of each equation contains an algebraic sum of the neighbouring field variables ($P_z|_{i+1/2,j+1/2} - P_z|_{i+1/2,j-1/2}$) in the grid. The neighbouring field variables are the field variables located at a distance of half a space step from the field component starting from ($O_x|_{i+1/2,j}$)

2.5.2 Spatial reciprocity

The discretization of the spatial derivatives, explained in Section 2.4, aimed at clarifying the relationship between the different equations and variables. In Fig. 2.8, the relationships emanating from (2.22) are illustrated more clearly. An arrow has been drawn starting from the location of the field variable for which the time derivative is present on the left hand side of an equation (this is called *the equation of that field variable*), each arrow points towards a neighbouring field variable used in the right hand side of (2.22). In Fig. 2.8 this is illustrated once for each field component.

- For the x -directed field component O_x at location $(i + 1/2, j)$.
- For the y -directed field component O_y at location $(i - 1, j + 1/2)$.
- For the z -directed field component P_z at location $(i - 1/2, j + 1/2)$.

When two neighbouring field variables are observed, it is important to emphasize that both field variables appear in each others equation. Moreover, the coefficient for a field variable in the equation of a neighbouring field variable is the opposite of the coefficient of this neighbouring field in the equation of the field variable, in the way it was written in (2.22). As an example consider:

$$\theta_r|_{i+1/2,j} \frac{dO_x|_{i+1/2,j}}{dt} = -\frac{c_0}{\Delta_x} P_z|_{i+1/2,j+1/2} + \dots \quad (2.23a)$$

and

$$\kappa_r|_{i+1/2,j+1/2} \frac{dP_z|_{i+1/2,j+1/2}}{dt} = \frac{c_0}{\Delta_x} O_x|_{i+1/2,j} + \dots \quad (2.23b)$$

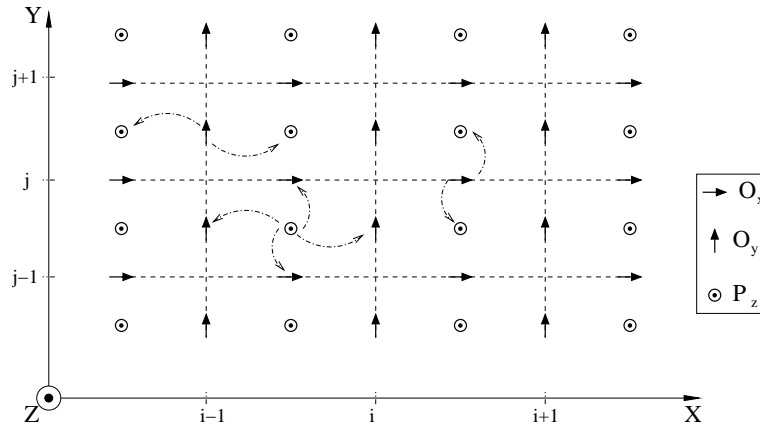


Figure 2.8: Each field component is related to its neighbours. Illustrated by the dashed-dotted arrow.

This property will be very important once the concept of state-space models is introduced and when stability is discussed. In [4], the term *spatial reciprocity*, as this property will be called here, was introduced for the first time.

2.5.3 Staircase approximation

The constitutive parameters, or the electromagnetic material properties ϵ_r for the electric field and μ_r for the magnetic field, have the value of the constitutive parameters of the material at the location of the field variables. The same holds for the losses σ_e and σ_m . When a field variable is located in the neighbourhood of two materials having different material properties, then the corresponding parameter is often set equal to the material at the field variable location. Consequentially this means that the material boundary can shift by almost a distance Δ without resulting in any changes in the problem being modeled. This is the price that has to be paid for the simplicity of the resulting spatially discretized equations (2.22).

Furthermore, when the boundary between two materials is a curved boundary, it will be approximated by a staircase-looking transition (see Fig. 2.9). This *staircase approximation* can, in most cases, be improved by using a finer grid, but this results in a higher number of field variables for the same problem space.

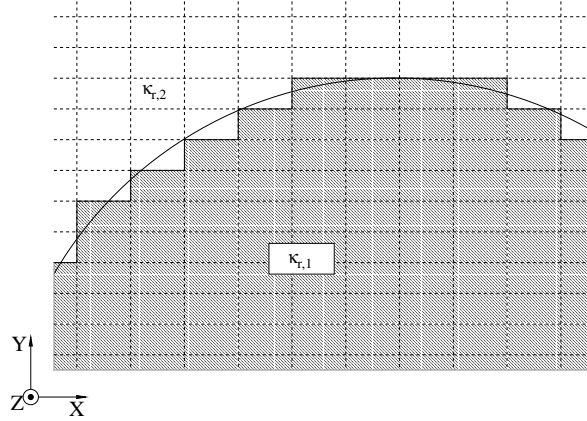


Figure 2.9: The full line represents the boundary between two materials with parameter $\kappa_{r,1}$ and $\kappa_{r,2}$. The staircase approximation of this curved boundary is illustrated.

2.5.4 Contour integral interpretation

Up to now, everything presented here was based on the differential form of Maxwell's equation. The same equations can be derived starting from Faraday's and Ampere's law in integral form. This is thoroughly investigated in [1] and in [5]. In this interpretation the average of a field component in an elementary cell is the field variable and not the value at that specific point. The same is true for the material parameters, in this way explaining the consequence of fitting a curved material boundary onto an orthogonal grid as shown in Fig. 2.9.

2.5.5 Square grid

In general, space can be discretized based on a different space step in both directions ($\Delta_x \neq \Delta_y$). From now on however we will restrict ourselves to the less general case of grids with square elementary cells (see Fig. 2.10), where $\Delta = \Delta_x = \Delta_y$. This will allow us to focus on the innovative aspects of the work presented here, without losing too much effort with aspects that are of less importance.

2.5.6 The space step

To be able to simulate all effects, the elementary cells have to be chosen small enough, thus allowing a higher density of field variables and obtain-

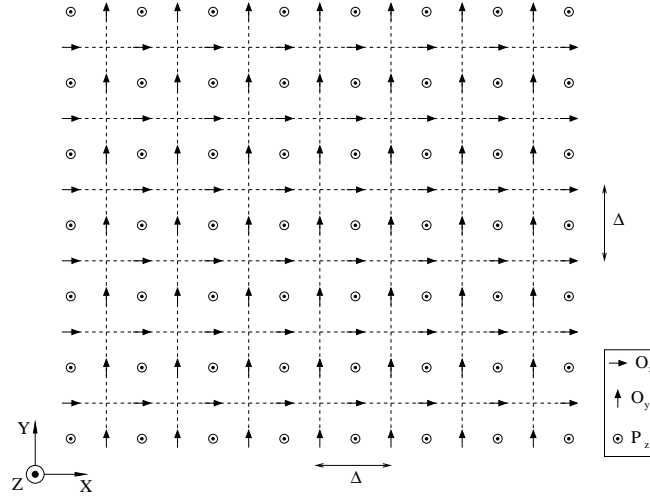


Figure 2.10: The regular uniform square grid and the location of the different field variables throughout the grid.

ing a better approximation of the continuous field behaviour. Since the field variables are spread uniformly over the grid in the original FDTD method, it is the smallest detail that will determine the space step and the size of the elementary cell. Two important conditions influence this choice:

- The smallest wavelength (λ_{\min}), corresponding to the highest frequency has to be discretized adequately. This requirement is unavoidable. As a good practice one assumes that Δ can be chosen between $\lambda_{\min}/10$ and $\lambda_{\min}/20$, making a compromise between accuracy (choosing Δ very small) and computation performance (choosing Δ large).
- Due to the fact that fields can vary rapidly near small features, it is often necessary to use a finer grid when small features have to be incorporated in simulations. To reasonably simulate these small objects the space step has to be even finer than these small objects, allowing thus to capture the possibly high variations of the fields.

Of course the final space step is the minimum of both requirements. Especially the requirement due to the small objects is tedious, since it obliges the user to change the global space step, although it is physically clear that this is only needed in the neighbourhood of the small object.

In this text we will discuss two existing techniques to alleviate this problem. The first one is called subgridding and will be discussed in the next section. It is based on the combination of two general FDTD grids, with

different space steps. The other approach that will be treated here is the so-called subcell model. In this approach, the known behaviour of the fields near a small geometric feature is used to adapt the corresponding FDTD equations. A short overview of existing subcell models will be given in Chapter 6. As we go along a new technique will be presented, the subdomain FDTD method, which is based on subgridding and can be interpreted as a technique to generate subcell models in an automatic and more general fashion.

3 SUBGRIDDING

3.1 Introduction

As explained at the end of the previous section, the simulation of fine geometrical features is difficult in a grid-based method as FDTD. One technique to tackle this problem is called *subgridding*. Subgridding means, the combination of two grids with a different space step. The first grid is called the *coarse grid*, and is used throughout the entire problem space. The second grid is called the *fine grid* and is based on a finer discretization. This fine grid is used locally around fine geometrical features that need to be incorporated in the simulation. Both grids are regular orthogonal grids as discussed in the previous section. The first problem that arises when combining two grids in one simulation, is how to connect both grids. All equations inside both grids are the same as in the standard FDTD method. Only the equations at the fine grid – coarse grid boundary have to be adapted, since certain neighbours are no longer available. A second problem arises when the time step used in both grids is not equal.

In this section we will only focus on the problem of different spatial discretization between both grids. As in the previous section, the discussion on time discretization will be postponed till the next chapter.

In a first part (subsection 3.2) a straightforward subgridding scheme will be introduced. This first attempt is based on breaking down coarse elementary cells into fine elementary cells. Afterwards in subsection 3.3, a more advanced subgridding scheme will allow the easy connection of the coarse grid to the fine grid. In subsection 3.4, a last adaptation of the grid will allow easy connection of the fine grid to the coarse grid. In a following part (subsection 3.5), the accuracy of the resulting equations will be investigated. There, the spatial reciprocity of the resulting mesh will also be under scrutiny. In a last part (subsection 3.6), a dual counterpart for the introduced grid will be presented.

3.2 Subgridding: first attempt

Based on the idea that a uniform orthogonal grid is merely a collection of identical elementary cells (see Fig. 2.2), a first proposition of subgridded mesh would be to double the number of cells per unit of length. This is shown in Figure 2.11. In the figure both the cells and the field variables of an interesting part of the grid have been represented. Normally the fine grid is entirely included in the coarse grid, in the figure only the corner region of the fine grid is shown.

As far as subgridding is concerned, often the space step related to the coarse grid (Δ_c) is a multiple of the fine grid space step (Δ_f). The ratio of both space steps (Δ_c/Δ_f) is called the *refinement ratio* (r), in the case of Fig 2.11 the refinement ratio is 2.

A similar grid with $r = 2$ was also used in [6], [7] and [8]. In [9] and [10], grids with $r = 4$ were presented. Most schemes presented in the literature are not easy to interpret, since the spatial and the temporal aspects of the subgridding method are explained together. Often, in addition to this, it is not possible to separate the spatial and the temporal discretization of the proposed methods since they rely on empirically changed algorithms [11], or the schemes show an overlap between the fine grid and the coarse grid [12] and [13]. In these subgridding methods this overlap is necessary to connect the fine grid and the coarse grid in the time stepping algorithm.

For all the field variables not located in the neighbourhood of the coarse grid – fine grid boundary, the equations are as (2.22), using Δ_c and Δ_f where appropriate. However, for the field variables located at the transition region between the coarse grid and the fine grid, it is no longer possible to use central differences to approximate the spatial derivatives, due to the fact that the regular grid is interrupted.

Let us first consider the field variables that correspond to the coarse grid and for which a neighbour, normally present in the coarse grid is missing. Yee's notation, where the same constant $\Delta_f = \Delta_c/2$ for field variables in both grids has been left out, is used. In Fig. 2.11, these field variables are $P_z|_{i+1,j+1}$, $P_z|_{i+3,j+2}$ and $P_z|_{i+5,j+2}$. These field variables will be called the *coarse grid boundary field variables* because they belong to the boundary of the coarse grid: these field variables are not solely surrounded by coarse grid field variables.

It is possible to artificially construct a missing neighbour field, by averaging the available fine grid fields. For instance for the equation of $P_z|_{i+1,j+1}$, we need $\partial O_y|_{i+1,j+1}/\partial x$, where the missing field $O_y|_{i+2,j+1}$ can

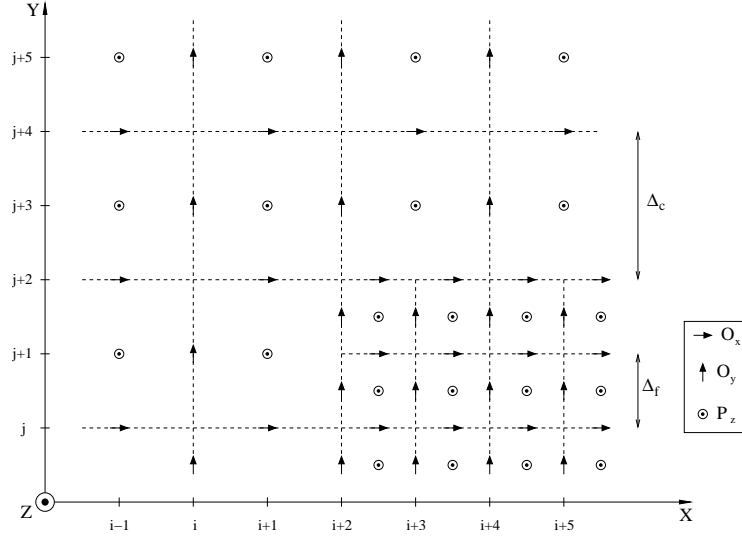


Figure 2.11: A first attempt to introduce a subgrid into a coarse grid. The refinement ratio is 2.

be replaced by an average:

$$\frac{\partial O_y}{\partial x} \Big|_{i+1,j+1} \approx \frac{\frac{O_y|_{i+2,j+3/2} + O_y|_{i+2,j+1/2}}{2} - O_y|_{i,j+1}}{2\Delta_f} \quad (2.24)$$

When using the Taylor series to expand a function $f(x, y)$ with two variables at the point $x = a, y = b$ [14]:

$$\begin{aligned} f(a+h, b+k) = f(a, b) &+ \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right) f(x, y) \Big|_{x=a, y=b} + \dots \\ &+ \frac{1}{n!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^n f(x, y) \Big|_{x=a, y=b} + \dots \end{aligned} \quad (2.25)$$

the accuracy of (2.24) can be determined. The Taylor series for the field variables at right hand side of (2.24) is:

$$\begin{aligned} O_y|_{i+2,j+3/2} &= O_y|_{i+1,j+1} + \Delta_f \left(\frac{\partial O_y|_{i+1,j+1}}{\partial x} + \frac{1}{2} \frac{\partial O_y|_{i+1,j+1}}{\partial y} \right) \\ &+ \frac{\Delta_f^2}{2} \left(\frac{\partial^2 O_y|_{i+1,j+1}}{\partial x^2} + \frac{\partial^2 O_y|_{i+1,j+1}}{\partial x \partial y} + \frac{1}{4} \frac{\partial^2 O_y|_{i+1,j+1}}{\partial y^2} \right) + \mathcal{O}(\Delta_f^3) \end{aligned} \quad (2.26a)$$

$$O_y|_{i+2,j+1/2} = O_y|_{i+1,j+1} + \Delta_f \left(\frac{\partial O_y|_{i+1,j+1}}{\partial x} - \frac{1}{2} \frac{\partial O_y|_{i+1,j+1}}{\partial y} \right) + \frac{\Delta_f^2}{2} \left(\frac{\partial^2 O_y|_{i+1,j+1}}{\partial x^2} - \frac{\partial^2 O_y|_{i+1,j+1}}{\partial x \partial y} + \frac{1}{4} \frac{\partial^2 O_y|_{i+1,j+1}}{\partial y^2} \right) + \mathcal{O}(\Delta_f^3) \quad (2.26b)$$

$$O_y|_{i,j+1} = O_y|_{i+1,j+1} - \Delta_f \frac{\partial O_y|_{i+1,j+1}}{\partial x} + \frac{\Delta_f^2}{2} \frac{\partial^2 O_y|_{i+2,j+1}}{\partial x^2} + \mathcal{O}(\Delta_f^3) \quad (2.26c)$$

resulting, for (2.24), in:

$$\begin{aligned} & \frac{\frac{O_y|_{i+2,j+3/2} + O_y|_{i+2,j+1/2}}{2} - O_y|_{i,j+1}}{2\Delta_f} \\ &= \frac{\partial O_y|_{i+1,j+1}}{\partial x} + \frac{\Delta_f}{16} \frac{\partial^2 O_y|_{i+1,j+1}}{\partial y^2} + \mathcal{O}(\Delta_f^2) \end{aligned} \quad (2.27)$$

Eqn. (2.27) indicates that the approximation is only first order accurate.

3.3 Odd refinement ratio

A way to avoid this, is to use a refinement ratio that is an odd number. A grid with refinement ratio $r = 3$ was introduced in [6], [15] and in [16]. Then the missing neighbours for the field variables related to the coarse grid are *colocated* with certain field variables related to the fine grid. Colocated means that the location for two field variables is exactly the same. The term colocated is frequently used in [11]. In Fig 2.12, this is illustrated. Take for instance $P_z|_{i+3/2,j+3/2}$. The x -directed derivative of O_y at that location can now be approximated by the normal second order accurate central difference approximation since in the fine grid the corresponding field variable ($O_y|_{i+3,j+3/2}$) is included.

Now consider the field variables located at the boundary of the fine grid. In Fig 2.12, these fields are:

$$O_y|_{i+3,j-1/2}, \dots, O_y|_{i+3,j+5/2} \quad (2.28)$$

and

$$O_x|_{i+7/2,j+3}, \dots, O_x|_{i+15/2,j+3} \quad (2.29)$$

These field variables will be called the *fine grid boundary field variables* since they belong to the boundary of the fine grid: these field variables are not solely surrounded by fine grid field variables. For each of these field variables, a neighbouring field, that would allow easy and accurate discretization of the spatial derivatives, is missing. Whereas for each missing neighbouring field of the field variables at the boundary of the coarse

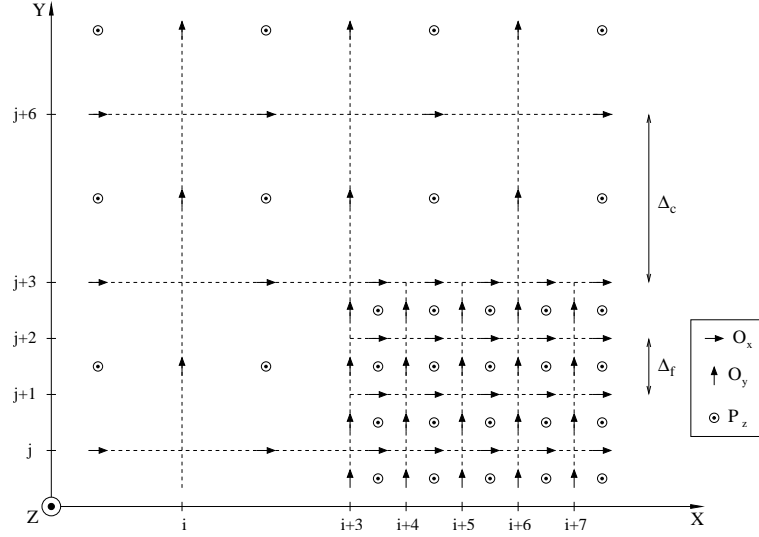


Figure 2.12: A coarse grid and a fine grid with an odd refinement ratio $r = 3$.

grid (e.g. neighbouring field of $P_z|_{i+3/2, j+3/2}$), several field variables at the boundary of the fine grid are present, this is not the case for the missing fields of the fine grid boundary field variables. This means that interpolation will be necessary in order to obtain an accurate approximation of the derivatives.

A logical approach would be to calculate the most accurate approximation of the spatial derivative using the available neighbouring field variables, e.g. for $O_x|_{i+1/2, j+3}$:

$$\left. \frac{\partial P_z}{\partial y} \right|_{i+1/2, j+3} \approx c_1 P_z|_{i+1/2, j+5/2} + c_2 P_z|_{i+9/2, j+9/2} + c_3 P_z|_{i+15/2, j+9/2} + \dots \quad (2.30)$$

where c_i $i = 1, \dots, n$ are constants calculated using (2.25). Unfortunately, this kind of approximation is not very elegant since it will oblige to incorporate a relatively high number — at least more than 2 — of neighbouring field variables to obtain the second order accuracy. These approximations are quite complicated in general since, in the direction for which the derivative has to be approximated (y -direction for $O_x|_{i+1/2, j+9/2}$) the distance of the neighbouring fields is $\Delta_f/2$ on the one side, and $\Delta_c/2$ on the other side.

Fortunately there is an easy way to adapt the grid allowing simple approximations of the derivatives to obtain a high accuracy. A first possibility

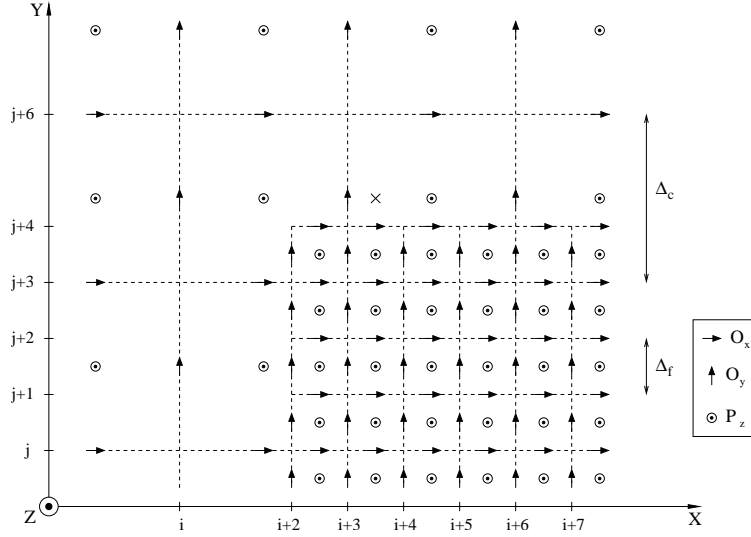


Figure 2.13: The combination of a coarse grid and a fine grid. The refinement ratio is: $r = 3$ and the fine grid region was expanded with 1 layer of cells.

would be to use $P_z|_{i+11/2, j+3/2}$ instead of $P_z|_{i+11/2, j+5/2}$. But there is a second possibility based on adapting the grid that will allow highly accurate approximations using simple interpolations.

3.4 Final grid

The boundary field variables of the fine grid, which are of the type O_x or O_y , can be brought closer to the boundary field variables of the coarse grid, which are of the type P_z by adding some layers of cells around the fine grid domain. In Fig. 2.13, the grid as shown in Fig. 2.12 was duplicated but one layer of fine cells was added around the fine grid. This allows the use of simple 1D interpolation to generate the missing neighbour that then can be used in a central difference approximation.

Say, for example, for the fine grid boundary field variable $O_x|_{i+7/2, j+4}$, the knowledge of the not existing field variable $P_z|_{i+7/2, j+9/2}$ is required, indicated in Fig. 2.13 with a \times -symbol. Conventional interpolation using the field variables $P_z|_{i+3/2, j+9/2}$ and $P_z|_{i+9/2, j+9/2}$ results in a first order approximation of the desired field variable. Using also $P_z|_{i+15/2, j+9/2}$ results in an approximation of second order of the desired field variable. The resulting

central difference approximation is then:

$$\left. \frac{\partial P_z}{\partial y} \right|_{i+7/2, j+4} \approx \frac{1}{\Delta_f} \left(\frac{2}{9} P_z|_{i+3/2, j+9/2} + \frac{8}{9} P_z|_{i+9/2, j+9/2} - \frac{1}{9} P_z|_{i+15/2, j+9/2} - P_z|_{i+7/2, j+7/2} \right) \quad (2.31)$$

The question whether this equation is fully second order accurate will be investigated hereafter (Section 3.5).

Adding some layers of fine cells is not only possible when $r = 3$ but can be done for every odd refinement ratio. However, the number of layers that needs to be added is in general $(r - 1)/2$. In this way the two dimensional interpolation problem for the boundary field variables has been reduced to an interpolation problem in only one dimension. The kind of grid shown in Fig. 2.13 has already been used: in [17], [18] a similar grid was proposed for a refinement ratio $r = 5$. It is also worth noting that the missing neighbours of the boundary coarse grid field variables still have collocated fine grid field variables.

3.5 Accuracy

Finally we will take a closer look at the resulting equations. For the fields at the coarse grid boundary, nothing changes since a fine grid field variable is present at the desired location. This implies that equation (2.22c) can be used, and this automatically has second order accuracy.

For the fine grid field variables, things are not so easy. Consider Fig. 2.14, which is a small part of the transition region between fine and coarse grid. The fine grid is situated at the bottom of the figure. Only four relevant field variables are shown: a field O_x at a certain location $(i + l, j)$ where $-r/2 \leq l \leq r/2$. At the location of the field variable also $\partial P_z / \partial y$ has to be approximated. The P_z field variables that will be used for the approximation have also been added. All other field variables have been left out.

First of all consider the Taylor series at the point $(i + l, j)$ for the different field variables:

$$\begin{aligned} P_z|_{i-r/2, j+1/2} = P_z|_{i+l, j} - \left(l + \frac{r}{2} \right) \Delta_f \left. \frac{\partial P_z}{\partial x} \right|_{i+l, j} + \frac{\Delta_f}{2} \left. \frac{\partial P_z}{\partial y} \right|_{i+l, j} \\ + \frac{\Delta_f^2}{2} \left[\left(l + \frac{r}{2} \right)^2 \left. \frac{\partial^2 P_z}{\partial x^2} \right|_{i+l, j} - \left(l + \frac{r}{2} \right) \left. \frac{\partial^2 P_z}{\partial x \partial y} \right|_{i+l, j} + \frac{1}{4} \left. \frac{\partial^2 P_z}{\partial y^2} \right|_{i+l, j} \right] \\ + \mathcal{O}(\Delta_f^3) \quad (2.32a) \end{aligned}$$

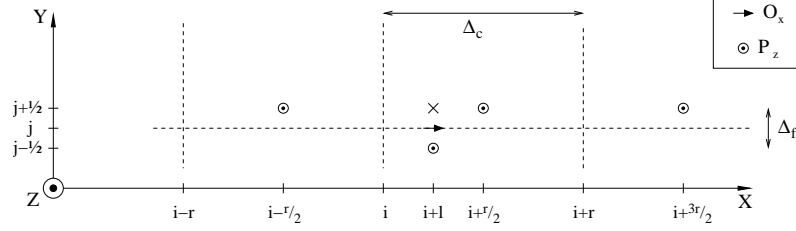


Figure 2.14: A small portion of the coarse grid - fine grid boundary region for the general case of refinement ratio r .

$$\begin{aligned}
 P_z|_{i+r/2, j+1/2} &= P_z|_{i+l, j} + \left(\frac{r}{2} - l\right) \Delta_f \frac{\partial P_z}{\partial x} \Big|_{i+l, j} + \frac{\Delta_f}{2} \frac{\partial P_z}{\partial y} \Big|_{i+l, j} \\
 &+ \frac{\Delta_f^2}{2} \left[\left(\frac{r}{2} - l\right)^2 \frac{\partial^2 P_z}{\partial x^2} \Big|_{i+l, j} + \left(\frac{r}{2} - l\right) \frac{\partial^2 P_z}{\partial x \partial y} \Big|_{i+l, j} + \frac{1}{4} \frac{\partial^2 P_z}{\partial y^2} \Big|_{i+l, j} \right] \\
 &+ \mathcal{O}(\Delta_f^3) \quad (2.32b)
 \end{aligned}$$

$$\begin{aligned}
 P_z|_{i+3r/2, j+1/2} &= P_z|_{i+l, j} + \left(\frac{3r}{2} - l\right) \Delta_f \frac{\partial P_z}{\partial x} \Big|_{i+l, j} + \frac{\Delta_f}{2} \frac{\partial P_z}{\partial y} \Big|_{i+l, j} \\
 &+ \frac{\Delta_f^2}{2} \left[\left(\frac{3r}{2} - l\right)^2 \frac{\partial^2 P_z}{\partial x^2} \Big|_{i+l, j} + \left(\frac{3r}{2} - l\right) \frac{\partial^2 P_z}{\partial x \partial y} \Big|_{i+l, j} + \frac{1}{4} \frac{\partial^2 P_z}{\partial y^2} \Big|_{i+l, j} \right] \\
 &+ \mathcal{O}(\Delta_f^3) \quad (2.32c)
 \end{aligned}$$

Calculating the weighted sum of these equations then yields:

$$\begin{aligned}
 \frac{3r^2 + 4l^2 - 8lr}{8r^2} P_z|_{i-r/2, j+1/2} + \frac{6r^2 - 8l^2 + 8lr}{8r^2} P_z|_{i+r/2, j+1/2} \\
 + \frac{-r^2 + 4l^2}{8r^2} P_z|_{i+3r/2, j+1/2} &= P_z|_{i+l, j} + \frac{\Delta_f}{2} \frac{\partial P_z}{\partial y} \Big|_{i+l, j} \\
 &+ \frac{\Delta_f^2}{4} \frac{\partial^2 P_z}{\partial y^2} \Big|_{i+l, j} + \mathcal{O}(\Delta_f^3) \quad (2.33)
 \end{aligned}$$

This result shows that second order accurate formulas are possible for this grid, by simply calculating a one dimensional interpolation along the straight line $y = (j + 1/2)\Delta_f$, for the location denoted by \times in Fig. 2.14. This becomes clear when (2.33) is used together with:

$$P_z|_{i+l, j-1/2} = P_z|_{i+l, j} - \frac{\Delta_f}{2} \frac{\partial P_z}{\partial y} \Big|_{i+l, j} + \frac{\Delta_f^2}{4} \frac{\partial^2 P_z}{\partial y^2} \Big|_{i+l, j} + \mathcal{O}(\Delta_f^3) \quad (2.34)$$

to calculate the second order accurate approximation of $\partial P_z / \partial y$ at $(i+l, j)$.

$$\begin{aligned} \frac{\partial P_z|_{i+l,j}}{\partial y} = & \frac{3r^2 - 8lr + 4l^2}{8r^2\Delta_f} P_z|_{i-r/2,j+1/2} + \frac{3r^2 + 4lr - 4l^2}{4r^2\Delta_f} P_z|_{i+r/2,j+1/2} \\ & + \frac{4l^2 - r^2}{8r^2\Delta_f} P_z|_{i+3r/2,j+1/2} - \frac{1}{\Delta_f} P_z|_{i+l,j-1/2} + \mathcal{O}(\Delta_f^2) \end{aligned} \quad (2.35)$$

The previous equations result in second order accurate approximations of the spatial derivative: the error is $\mathcal{O}(\Delta_f^2)$. In this way the assumption we made in the previous section has been confirmed. This kind of interpolation can be performed for every location of the field variable, as long as in the direction of the derivative it is central ($y = j\Delta_f$) between its neighbours ($y = (j+1/2)\Delta_f$ and $y = (j-1/2)\Delta_f$). So it can certainly be performed for grids constructed as in Section 3.4, where the refinement ratio is an odd number.

Starting from the same grid, a more simple linear interpolation can be used as well. Then the desired field variable is constructed as:

$$\begin{aligned} \left(\frac{1}{2} + \frac{l}{r}\right) P_z|_{i+r/2,j+1/2} + \left(\frac{1}{2} - \frac{l}{r}\right) P_z|_{i-r/2,j+1/2} \\ = P_z|_{i+l,j} + \frac{\Delta_f}{2} \frac{\partial P_z}{\partial y} \Big|_{i+l,j} + \mathcal{O}(\Delta_f^2) \end{aligned} \quad (2.36)$$

However the resulting approximation of the derivative will only be first order accurate.

An important remark at this point is the loss of spatial reciprocity, as it was introduced in Section 2.5.2, and this especially at the fine grid – coarse grid boundary. Remember, spatial reciprocity would mean that if in the equation of a certain field variable an other field variable is used, then in the equation of that field variable the concerned field variable is used. In Fig. 2.15 this is graphically illustrated for one fine grid boundary field variable and for one coarse grid boundary field variable. The arrows arriving at the concerned field variable indicate in which equations it is used. The arrows leaving the concerned field variable indicate the field variables used in the equation of that field variable. This clearly visualizes the loss of reciprocity, since only few arrows form a loop. Or graphically with the dashed-dotted arrows spatial reciprocity means that every arrow is accompanied by an arrow in the other direction. In Fig. 2.15 the refinement ratio was $r = 3$. Similar results can be obtained for higher refinement ratios. The arrows are related to the first order accurate approximation of the fine grid boundary field variables (2.36). The second order accurate approximations would just add more arrows without a returning counterpart.

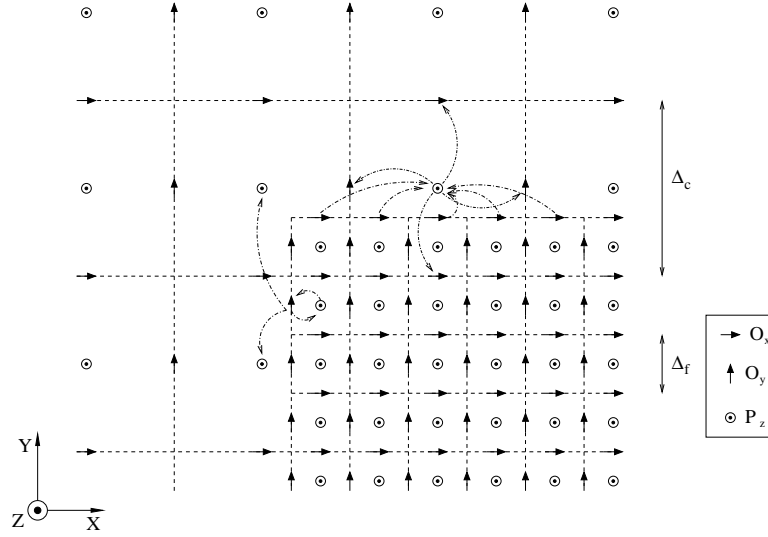


Figure 2.15: The combination of a fine grid and a coarse grid with $r = 3$, illustrating the relationship of boundary field variables and their neighbours.

Up to now, the main concern was to propose an accurate subgridding scheme. In this way reciprocity was lost out of sight. As will be explained later, spatial reciprocity is important when stability of finite difference algorithms is concerned. In the literature some schemes have been presented that preserve this spatial reciprocity. The first paper concerning this subject was published by Thoma et al in [7]. Around the same time similar ideas were proposed by Krishnaiah et al in [16] and [15]. It is believed that when subgridding is involved, both properties (accuracy and spatial reciprocity) are difficult to combine.

3.6 Dual grid

The subgridding technique as illustrated in Fig. 2.13 has the following properties:

- coarse grid boundary fields are of the type P_z
- fine grid boundary fields are of the type O : either O_x or O_y
- overall second order accurate approximations are possible

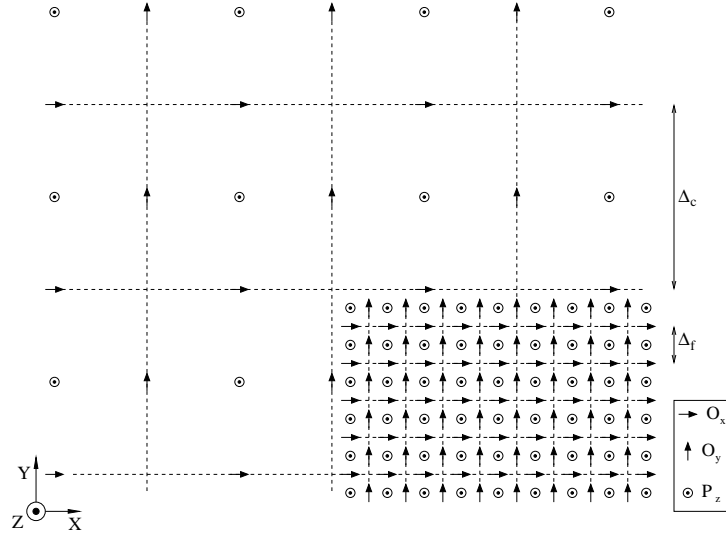


Figure 2.16: The dual situation for combining a fine grid and a coarse grid in 2D, $r = 5$.

There is a dual situation for this kind of grid, where fine grid boundary field variables are P_z and where the coarse grid boundary field variables are either O_x or O_y . This dual grid has the same accuracy properties. In Fig. 2.16 it is illustrated for the case of $r = 5$.

4 STATE-SPACE MODELS

4.1 Introduction

In control systems theory, systems can often be described by means of a number of time-invariant first-order differential equations. When for this linear, time-invariant system input variables and output variables can be defined, then this system can be poured into a state-space representation. In circuit theory, for instance, circuits are often described in this way. The input and output variables are then the voltages and the currents at a limited number of ports. The modified nodal analysis method [19] is but one example of such a circuit-equation formulation method.

In Section 4.2 some general aspects of state-space representations will be introduced. Afterwards, in Section 4.3, based on the principles set forth in the previous section, a state-space description of the field variables of a

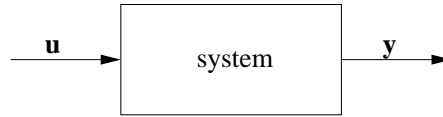


Figure 2.17: The block representation of a simple system.

finite difference grid will be introduced. This will be done not only for the uniform orthogonal grid but also for the combination of a fine grid and a coarse grid. Finally, a closer look will be taken at the implications of tearing up a grid into several subdomains.

4.2 General aspects

A state-space model is a set of first-order differential equations. It is used to represent a linear, time-invariant system. By remarking that an n^{th} -order differential equation can be written as a system of n first order differential equations, it follows that these systems are of the following form:

$$\begin{cases} \mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{L}^T\mathbf{x} \end{cases} \quad (2.37)$$

A block representation of this system is shown in Fig. 2.17. This illustrates the conceptual ideas of a state-space model, namely a black-box model containing the state of the system, steered by some input \mathbf{u} and monitored by some output \mathbf{y} . The first equation of (2.37) expresses how the state of the system (\mathbf{x}) evolves in time. In [20], the definition of the state of a system is given as follows:

The state of a system is the minimum set of numbers or variables, the state variables, which contain sufficient information about the past history of the system to permit us to compute all future states of the system — assuming, of course that all future inputs (control forces) are known and also the equations (bonds of interactions) describing the system.

The second equation of (2.37) expresses how the system output variables contained in the vector \mathbf{y} can be extracted from the state of the system \mathbf{x} , or also referred to as the internal variables.

In equation (2.37) several vectors can be observed. First of all, the vector \mathbf{u} , containing the systems *input variables* can be seen. Second the vector \mathbf{y} , this vector contains the *output variables* of the system. The idea of

input and output variables is more clearly illustrated in Fig. 2.17. The input and output variables express how the system can communicate with other systems. The dimension of \mathbf{u} and \mathbf{y} can be one, then the system is a SISO-system (single-input/single-output), or larger ($\dim[\mathbf{u}] = p$, $\dim[\mathbf{y}] = m$) for a MIMO-system (multiple-input/multiple-output). Third, the vector \mathbf{x} can be distinguished. This vector contains the internal variables of the system. Equation (2.37) not only contains \mathbf{x} , but also contains $\dot{\mathbf{x}} = d\mathbf{x}/dt$. In general the dimension of \mathbf{x} ($\dim[\mathbf{x}] = N$) is considerably larger than the dimension of either \mathbf{u} or \mathbf{y} .

The linear relationship between these variables is expressed by the matrices \mathbf{C} , \mathbf{G} , \mathbf{B} and \mathbf{L} . The dimensions of these matrices correspond to the vectors: \mathbf{C} and \mathbf{G} are of dimension $N \times N$. The *input matrix* \mathbf{B} is of dimension $p \times N$ and the *output matrix* \mathbf{L} is of dimension $m \times N$.

4.3 Finite difference grids

4.3.1 The uniform orthogonal grid

Let us start by reconsidering the discretized Maxwell's equations, where $\Delta_x = \Delta_y = \Delta$:

$$\theta_r|_{i+1/2,j} \frac{d O_x|_{i+1/2,j}}{dt} = -c_0 \frac{P_z|_{i+1/2,j+1/2} - P_z|_{i+1/2,j-1/2}}{\Delta} - \sigma|_{i+1/2,j} O_x|_{i+1/2,j} \quad (2.22a)$$

$$\theta_r|_{i,j+1/2} \frac{d O_y|_{i,j+1/2}}{dt} = c_0 \frac{P_z|_{i+1/2,j+1/2} - P_z|_{i-1/2,j+1/2}}{\Delta} - \sigma|_{i,j+1/2} O_y|_{i,j+1/2} \quad (2.22b)$$

$$\kappa_r|_{i+1/2,j+1/2} \frac{d P_z|_{i+1/2,j+1/2}}{dt} = c_0 \left[\frac{O_y|_{i+1,j+1/2} - O_y|_{i,j+1/2}}{\Delta} - \frac{O_x|_{i+1/2,j+1} - O_x|_{i+1/2,j}}{\Delta} \right] - \sigma^*|_{i+1/2,j+1/2} P_z|_{i+1/2,j+1/2} \quad (2.22c)$$

With each field variable present in a grid, an equation of this form is associated. This results in a, often very large, set of first-order differential equations. When all these equations are written down in matrix form, we end up with a state-space model of the form (2.37). The flexibility of this notation allows the definition of several types of input and output variables. Two different types of input and output variables have to be distinguished:

1. The first type is for a state-space model representing all field variables of an entire grid. There the input variables are the sources or

the excitations of the corresponding FDTD-problem and the output variables are the resulting physical quantities that we want to know. A typical problem is a waveguide problem, where the input variable is the excitation of the inserted mode. The output variable is then a field variable that allows the recording of a propagating mode. An other example is the simulation of a microstrip line. There the input variable is a voltage that is being imposed. This problem has a single input, but it influences several field variables. The output variable can then be the algebraic sum of a well selected set of magnetic fields (i.e. representing a contour around the microstrip line), equivalent to the current passing through the microstrip line. In this section this type of state-space models will be treated, however we will not go into the detailed nature of the input and output variables, since these are problem dependent. The focus will be on the $\mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x}$ part of equation (2.37).

2. It is not necessary, though it is the most logical choice, to associate one state-space model to one grid. It is possible to cut a grid up into several subdomains. For each subdomain, a state-space model can then be written down. For a subdomain and the associated state-space model, the input variables are then the missing neighbours of that subdomain. These input variables all have to come from other subdomains. In this way the output variables of a subdomain are the field variables needed by field variables not present in that subdomain. The input and output variables of a state-space model then describe the interaction between the different subdomains. These kinds of state-space models will be the subject of the next section.

Of course the input or output variables can be the combination of both.

4.3.1.1 Boundary conditions

When a grid is under consideration this grid has to be terminated in one way or the other. For open domain problems treated with the FDTD method several techniques have been formulated to attach absorbing boundary conditions to the grid. Two of the most important absorbing boundary conditions are: the Mur boundary condition [21] and the perfectly matched layer (PML) [22]. Yet here, for theoretical reasons that will become clear further on, we will often suppose the grid to be surrounded by perfect electric conductors: the tangential electric field equals zero. Or surrounded by, mathematically interesting, perfect magnetic conductors: the tangential magnetic field equals zero. The consequence of terminating a grid with zero fields is that all the equations forming the state-space model are of

the form (2.22), except near the termination of the grid where some neighbouring field variables can be left out since they are zero.

4.3.1.2 Aligning equations and field variables

For a state-space model associated with a grid, a first choice is to number the equations according to the organization of the field variables inside \mathbf{x} . Remember: the equation associated with a field variable is the equation where the time derivative of that field variable is present. A consequence of this choice is that the matrix \mathbf{C} will be a diagonal matrix, where:

$$c_{kk} = \begin{cases} \theta_r|_{i+1/2,j} & \text{for } x_k = O_x|_{i+1/2,j} \\ \theta_r|_{i,j+1/2} & \text{for } x_k = O_y|_{i,j+1/2} \\ \kappa_r|_{i+1/2,j+1/2} & \text{for } x_k = P_z|_{i+1/2,j+1/2} \end{cases} \quad (2.38)$$

where the relative constitutive values form the diagonal elements, consequently:

$$c_{kk} \geq 1 \quad \forall k = 1, \dots, N \quad (2.39)$$

and matrix \mathbf{C} is positive definite: $\mathbf{y}^H \mathbf{C} \mathbf{y} > 0$ for each \mathbf{y} , where \mathbf{y}^H is the hermitian of vector \mathbf{y} . When the equations would not have been chosen in the same order as used for the field variables in the vector \mathbf{x} , the resulting state-space model would merely be a permutation of the former one. This would obscure some properties of the matrices.

4.3.1.3 Ordering \mathbf{x}

The ordering of the field variables in \mathbf{x} determines the structure of the matrices \mathbf{G} and \mathbf{C} . Here, first all the x -directed fields (O_x -type) will be inserted in \mathbf{x} , then all the y -directed fields (O_y -type) and finally all the z -directed fields (P_z -type), so:

$$\mathbf{x} = \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{p}_z \end{bmatrix} \quad (2.40)$$

The O_x -field variables inside \mathbf{o}_x are ordered by their coordinates: smaller x -coordinates come first, for each x -coordinate the field variables are ordered by the y -coordinate. The vectors \mathbf{o}_y and \mathbf{p}_z are ordered in the same manner. Introducing as well \mathbf{o} and \mathbf{p} :

$$\mathbf{o} = \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \end{bmatrix} \quad \mathbf{p} = \mathbf{p}_z \quad (2.41)$$

\mathbf{x} becomes:

$$\mathbf{x} = \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (2.42)$$

and the matrices can be written in the following block notation:

$$\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (2.43)$$

Since the \mathbf{C} -matrix (see (2.38)) is diagonal, \mathbf{C} can be written as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \quad (2.44)$$

where every \mathbf{D} -matrix is diagonal. The matrix \mathbf{D}_θ contains the constitutive parameters of the O_x and O_y fields. The matrix \mathbf{D}_κ contains the constitutive parameters of the P_z fields.

The coefficients on the right hand side of (2.22), form the elements of the matrix \mathbf{G} . In each equation, a loss factor is attached to the corresponding field variable, these form the elements of \mathbf{G}_{11} and \mathbf{G}_{22} . All other coefficients are elements of \mathbf{G}_{12} and \mathbf{G}_{21} , since these coefficients relate O_x or O_y field variables to P_z field variables, or vice versa. At this point the spatial reciprocity of the uniform orthogonal grid can be exploited in this matrix formulation. Say $x_k = O_x|_{i+1/2,j}$ and $x_l = P_z|_{i+1/2,j+1/2}$, then $[\mathbf{G}]_{kl} = -c_0/\Delta$. But considering again (2.23), then $[\mathbf{G}]_{lk} = c_0/\Delta$. This property is valid for each field variable, so $\mathbf{G}_{12} = -\mathbf{G}_{21}^T$. Since each non-zero element of \mathbf{G}_{12} is either c_0/Δ or $-c_0/\Delta$, it is possible to write:

$$\mathbf{G}_{12} = -\mathbf{G}_{21}^T = \frac{c_0}{\Delta} \mathbf{K} \quad (2.45)$$

where the non-zero elements of the matrix \mathbf{K} are either 1 or -1 . The matrices \mathbf{G}_{12} , \mathbf{G}_{21} and here \mathbf{K} can be interpreted as the discretized curl operator related to the grid that is used for the discretization, since these blocks relate the O -type fields to the P -type fields.

Summarized, the following is obtained:

$$\mathbf{G} = \begin{bmatrix} \mathbf{D}_\sigma & \frac{c_0}{\Delta} \mathbf{K} \\ -\frac{c_0}{\Delta} \mathbf{K}^T & \mathbf{D}_{\sigma^*} \end{bmatrix} \quad (2.46)$$

Again \mathbf{D}_σ and \mathbf{D}_{σ^*} are diagonal containing the loss parameters, σ and σ^* respectively, belonging to the different field variables. Since $\sigma, \sigma^* \geq 0$, both submatrices are positive semi-definite, $\mathbf{y}^H \mathbf{D}_\sigma \mathbf{y} \geq 0$ for each \mathbf{y} .

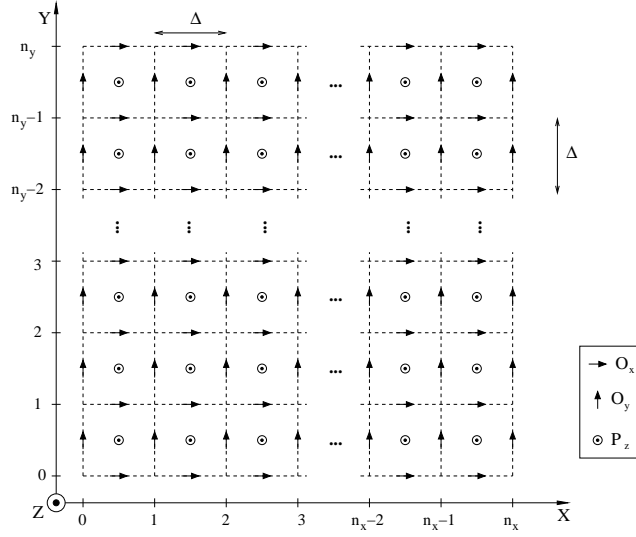


Figure 2.18: A regular orthogonal square grid of size $n_x \times n_y$ cells. The grid is terminated by perfect conducting walls: $P_z = 0$.

4.3.1.4 Explicit form for \mathbf{K} when at the boundary $P_z = 0$

In Fig. 2.18 a typical orthogonal regular grid is shown. The grid is terminated by a perfect conducting box where $P_z = 0$. The boundaries are located at:

$$x = -\frac{\Delta}{2} \quad x = \left(n_x + \frac{1}{2}\right) \Delta \quad (2.47)$$

and at

$$y = -\frac{\Delta}{2} \quad y = \left(n_y + \frac{1}{2}\right) \Delta \quad (2.48)$$

The choices, in order of importance, that were made to order the field variables in the vector \mathbf{x} are as explained in the previous paragraph:

1. according to orientation of the fields: first x -oriented, then y -oriented and finally the z -oriented
2. according to the x -coordinate: lower x -coordinates come before higher coordinates
3. according to the y -coordinate: lower y -coordinates come before higher coordinates

These choices allow us to write the position of each field variable inside the vector \mathbf{x} , namely:

$$O_x|_{i+1/2,j} = \mathbf{x}_{i(n_y+1)+j+1} \quad (2.49)$$

$$O_y|_{i,j+1/2} = \mathbf{x}_{n_x(n_y+1)+in_y+j+1} \quad (2.50)$$

$$P_z|_{i+1/2,j+1/2} = \mathbf{x}_{n_x(n_y+1)+(n_x+1)n_y+in_y+j+1} \quad (2.51)$$

To make things clearer, it is useful to write out the vector \mathbf{x} in more detail.

1. The first choice for ordering the field variables allows us to write:

$$\mathbf{x} = \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{p}_z \end{bmatrix} \quad (2.52)$$

2. The second choice allows us to write, for example, for \mathbf{o}_x

$$\mathbf{o}_x = \begin{bmatrix} \mathbf{o}_x|_{1/2} \\ \mathbf{o}_x|_{3/2} \\ \vdots \\ \mathbf{o}_x|_{n_x-1/2} \end{bmatrix} \quad (2.53)$$

where every vector $\mathbf{o}_x|_{i+1/2}$ contains the field variables O_x located at $x = (i + 1/2)\Delta$. The vectors \mathbf{o}_y and \mathbf{p}_z are then:

$$\mathbf{o}_y = \begin{bmatrix} \mathbf{o}_y|_0 \\ \mathbf{o}_y|_1 \\ \vdots \\ \mathbf{o}_y|_{n_x} \end{bmatrix} \quad \mathbf{p}_z = \begin{bmatrix} \mathbf{p}_z|_{1/2} \\ \mathbf{p}_z|_{3/2} \\ \vdots \\ \mathbf{p}_z|_{n_x-1/2} \end{bmatrix} \quad (2.54)$$

3. The third choice allows us to write, e.g. for $\mathbf{o}_x|_{i+1/2}$:

$$\mathbf{o}_x|_{i+1/2} = \begin{bmatrix} O_x|_{i+1/2,0} \\ O_x|_{i+1/2,1} \\ \vdots \\ O_x|_{i+1/2,n_y} \end{bmatrix} \quad (2.55)$$

the vectors \mathbf{o}_y and \mathbf{p}_z are then:

$$\mathbf{o}_y|_i = \begin{bmatrix} O_y|_{i,1/2} \\ O_y|_{i,3/2} \\ \vdots \\ O_y|_{i,n_y-1/2} \end{bmatrix} \quad \mathbf{p}_z|_{i+1/2} = \begin{bmatrix} P_z|_{i+1/2,1/2} \\ P_z|_{i+1/2,3/2} \\ \vdots \\ P_z|_{i+1/2,n_y-1/2} \end{bmatrix} \quad (2.56)$$

Making use of (2.22c), expressing how every P_z field variable is related to its four neighbours and the way the field variables are elements of \mathbf{x} , we can clarify the structure of \mathbf{K}^T and therefore of \mathbf{K} . It is possible to divide the \mathbf{K} -matrix into two parts. The first part expresses the relation between the P_z -type fields and the O_x -type fields whereas the second part yields the relationship between the P_z -type fields and the O_y -type fields, thus:

$$\mathbf{D}_\kappa \dot{\mathbf{p}}_z = \frac{c_0}{\Delta} \mathbf{K}^T \mathbf{o} - \mathbf{D}_{\sigma^*} \mathbf{p}_z \quad (2.57)$$

$$= \frac{c_0}{\Delta} \begin{bmatrix} \mathbf{K}_x^T & \mathbf{K}_y^T \end{bmatrix} \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \end{bmatrix} - \mathbf{D}_{\sigma^*} \mathbf{p}_z \quad (2.58)$$

As a first step we examine \mathbf{K}_x^T , representing the relationship between \mathbf{o}_x and \mathbf{p}_z , leading to the following observations:

- the field variables contained in $\mathbf{p}_z|_{i+1/2}$ are, as far as O_x -type field variables are concerned, only related to field variables contained in $\mathbf{o}_x|_{i+1/2}$.
- the relationship between $\mathbf{p}_z|_{i+1/2}$ and $\mathbf{o}_x|_{i+1/2}$ is identical for every i and is represented by:

$$\frac{c_0}{\Delta} \mathbf{W}_{n_y}^T = \frac{c_0}{\Delta} \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & -1 & \\ & & & & 1 & -1 \\ & & & & & 1 & -1 \end{bmatrix} \quad (2.59)$$

where $\mathbf{W}_{n_y}^T \in \mathbb{R}^{n_y \times (n_y+1)}$ and where a matrix \mathbf{W}_s is fully determined by the index s :

$$[\mathbf{W}_s]_{kl} = \begin{cases} 1 & \text{if } k = l \\ -1 & \text{if } k + 1 = l \\ 0 & \text{else} \end{cases} \quad \text{and } \mathbf{W}_s \in \mathbb{R}^{(s+1) \times s} \quad (2.60)$$

The total matrix \mathbf{K}_x^T can then be written as:

$$\mathbf{K}_x^T = \begin{bmatrix} \mathbf{W}_{n_y}^T & & & \\ & \mathbf{W}_{n_y}^T & & \\ & & \ddots & \\ & & & \mathbf{W}_{n_y}^T \end{bmatrix} \quad (2.61)$$

$$= \mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y}^T \quad (2.62)$$

where \mathbf{I}_r is the identity matrix belonging to $\mathbb{R}^{r \times r}$ and where \otimes denotes the *Kronecker product* [23]. For two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$, the Kronecker product of these two matrices is given by:

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{mp \times nq} \quad (2.63)$$

In a second step we examine \mathbf{K}_y^T , giving the relationship between \mathbf{o}_y and \mathbf{p}_z . Similar to \mathbf{K}_x^T the following observations can be made:

- the field variables contained in $\mathbf{p}_z|_{i+1/2}$ are only related to the field variables related in $\mathbf{o}_y|_i$ and the field variables in $\mathbf{o}_y|_{i+1}$.
- the relationship between $\mathbf{p}_z|_{i+1/2}$ and $\mathbf{o}_y|_i$ is identical for every i and opposite to the relationship between $\mathbf{p}_z|_{i+1/2}$ and $\mathbf{o}_y|_{i+1}$.

So this enables us to write, making use of (2.60):

$$\mathbf{K}_y^T = \begin{bmatrix} -\mathbf{I}_{n_y} & \mathbf{I}_{n_y} & & & \\ & -\mathbf{I}_{n_y} & \mathbf{I}_{n_y} & & \\ & & -\mathbf{I}_{n_y} & \ddots & \\ & & & \ddots & \mathbf{I}_{n_y} \\ & & & & -\mathbf{I}_{n_y} & \mathbf{I}_{n_y} \end{bmatrix} \quad (2.64)$$

$$= -\mathbf{W}_{n_x}^T \otimes \mathbf{I}_{n_y} \quad (2.65)$$

Finally, the matrix \mathbf{K}^T can be written in the form:

$$\mathbf{K}^T = \begin{bmatrix} \mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y}^T & -\mathbf{W}_{n_x}^T \otimes \mathbf{I}_{n_y} \end{bmatrix} \quad (2.66)$$

and exploiting the following Kronecker product property:

$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T \quad (2.67)$$

this gives for \mathbf{K} :

$$\mathbf{K} = \begin{bmatrix} \mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y} \\ -\mathbf{W}_{n_x} \otimes \mathbf{I}_{n_y} \end{bmatrix} \quad (2.68)$$

4.3.1.5 Explicit form of \mathbf{K} when at the boundary tangential O fields are zero

In Fig. 2.19 a rectangular grid is shown. However this time the grid is terminated by a dual boundary condition: the tangential O -type fields at

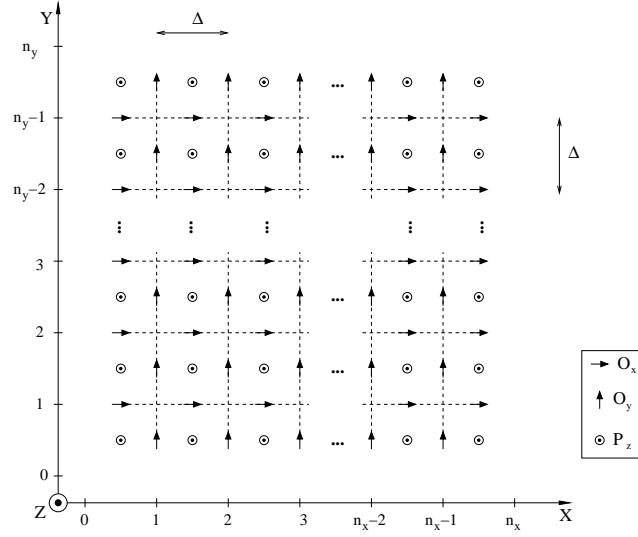


Figure 2.19: A regular orthogonal square grid. The grid is terminated by zero tangential fields at $x = 0$, $x = n_x\Delta$, $y = 0$ and $y = n_y\Delta$.

the boundary are zero. The grid is terminated by a perfect conducting box where $O_x = 0$ at the boundaries:

$$y = 0 \quad y = n_y\Delta \quad (2.69)$$

and where $O_y = 0$ at the boundaries:

$$x = 0 \quad x = n_x\Delta \quad (2.70)$$

Based on the same principles for ordering the field variables in \mathbf{x} , similar results as in the previous paragraph can be derived:

$$O_x|_{i+1/2,j} = x_{i(n_y-1)+j} \quad (2.71)$$

$$O_y|_{i,j+1/2} = x_{n_x(n_y-1)+(i-1)n_y+j+1} \quad (2.72)$$

$$P_z|_{i+1/2,j+1/2} = x_{n_x(n_y-1)+(n_x-1)n_y+in_y+j+1} \quad (2.73)$$

and

$$\mathbf{K} = \begin{bmatrix} -\mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y-1}^T \\ \mathbf{W}_{n_x-1}^T \otimes \mathbf{I}_{n_y} \end{bmatrix} \quad (2.74)$$

4.3.1.6 Sparsity of \mathbf{C} and \mathbf{G}

It has to be remarked that the matrices \mathbf{C} and \mathbf{G} are very sparse. For the matrix \mathbf{C} this is clear since it is a diagonal matrix. The matrix \mathbf{G} is also very

sparse, since each row has one element on the diagonal and in addition to this 2 (for each O_x and O_y field) or 4 (for each P_z field) other non-zero elements.

4.3.2 Subdomain

4.3.2.1 Graphical interpretation

As stated before, not only an entire grid can be described in state-space terminology, but this is also possible for a part of a grid: a *subdomain*. In Fig. 2.20 a small part of a grid is shown. There, a possible cut C has been introduced. The cut splits the grid up into two different parts: the field variables located at the left side of the cut (subdomain 1) and the field variables located at the right side of the cut (subdomain 2). Clearly all field variables present in the grid are either part of subdomain 1 or of subdomain 2. For both subdomains it is possible to write them as a state-space model. For subdomain 1, this is:

$$\begin{cases} \mathbf{C}_1 \dot{\mathbf{x}}_1 = -\mathbf{G}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u}_1 \\ \mathbf{y}_1 = \mathbf{L}_1^T \mathbf{x}_1 \end{cases} \quad (2.75a)$$

and for subdomain 2:

$$\begin{cases} \mathbf{C}_2 \dot{\mathbf{x}}_2 = -\mathbf{G}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u}_2 \\ \mathbf{y}_2 = \mathbf{L}_2^T \mathbf{x}_2 \end{cases} \quad (2.75b)$$

As mentioned before the interpretation of input variables and output variables for each state-space model has now changed. These variables are no longer related to the physical quantities that need to be inserted into the grid as source or extracted as result of the simulation, but they perform the communication between the different subdomains. This is graphically illustrated in Fig. 2.20.

Not for all field variables present in a certain subdomain the equations can be written down in the form $\mathbf{C}_\alpha \dot{\mathbf{x}}_\alpha = -\mathbf{G}_\alpha \mathbf{x}_\alpha$. Take for instance subdomain 1. The field variables denoted by a box (\square) around the field variable, have neighbouring field variables, denoted by a triangle (∇) that are located in the other subdomain. The field variables needed to complete the set of equations are all grouped in \mathbf{u}_1 , containing the input variables of state-space model 1 (2.75a).

Vice versa, some field variables in subdomain 2 (denoted by ∇), have neighbouring field variables (denoted by \square) located in subdomain 1. Again these missing field variables can be grouped into \mathbf{u}_2 . The matrix \mathbf{L}_1 allows the selection of the appropriate field variables from \mathbf{x}_1 . These field variables are then grouped in the vector \mathbf{y}_1 . It must be clear by now that the

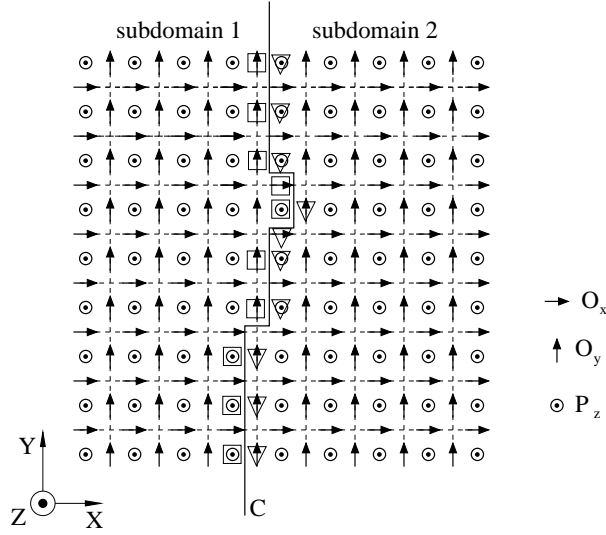


Figure 2.20: A typical grid where a cut C splits up the grid into 2 subdomains.

output variables of subdomain 1 are selected in such a way that they can act as the input variables of subdomain 2. Of course the same applies with respect to the relation between the output variables of subdomain 2 and the input variables of subdomain 1. In summary:

$$\mathbf{u}_1 = \mathbf{y}_2 \quad (2.76a)$$

$$\mathbf{u}_2 = \mathbf{y}_1 \quad (2.76b)$$

The two state-space models (2.75) are then clearly linked to each other. The block presentation for the combination of both state-space models is as shown in Fig. 2.21.

When the field variables in \mathbf{x}_1 and in \mathbf{x}_2 are ordered according to their type:

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{p}_1 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} \mathbf{o}_2 \\ \mathbf{p}_2 \end{bmatrix} \quad (2.77)$$

it is possible to write (2.75) in the following block notation:

$$\begin{cases} \begin{bmatrix} \mathbf{D}_{\theta,1} & 0 \\ 0 & \mathbf{D}_{\kappa,1} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}}_1 \\ \dot{\mathbf{p}}_1 \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma,1} & \frac{c_0}{\Delta} \mathbf{K}_1 \\ -\frac{c_0}{\Delta} \mathbf{K}_1^T & \mathbf{D}_{\sigma^*,1} \end{bmatrix} \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{p}_1 \end{bmatrix} + \mathbf{B}_1 \mathbf{L}_2^T \begin{bmatrix} \mathbf{o}_2 \\ \mathbf{p}_2 \end{bmatrix} \\ \begin{bmatrix} \mathbf{D}_{\theta,2} & 0 \\ 0 & \mathbf{D}_{\kappa,2} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}}_2 \\ \dot{\mathbf{p}}_2 \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma,2} & \frac{c_0}{\Delta} \mathbf{K}_2 \\ -\frac{c_0}{\Delta} \mathbf{K}_2^T & \mathbf{D}_{\sigma^*,2} \end{bmatrix} \begin{bmatrix} \mathbf{o}_2 \\ \mathbf{p}_2 \end{bmatrix} + \mathbf{B}_2 \mathbf{L}_1^T \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{p}_1 \end{bmatrix} \end{cases} \quad (2.78)$$

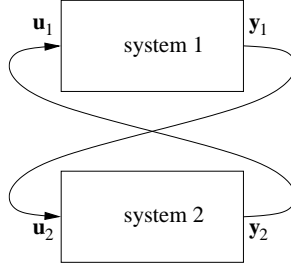


Figure 2.21: The block representation of the combined system (2.75).

where equation (2.76) was explicitly used. Again the \mathbf{D} -matrices are diagonal. Within each subdomain, the spatial reciprocity reappears. But there is more. When $\mathbf{B}_1 \mathbf{L}_2^T$, which gives the relationship between the field variables in subdomain 1 and the field variables in subdomain 2, is examined more in detail, it is clear that in $\mathbf{B}_1 \mathbf{L}_2^T$ the blocks relating $\dot{\mathbf{o}}_1$ to \mathbf{o}_2 and $\dot{\mathbf{p}}_1$ to \mathbf{p}_2 are zero. Therefore we can write:

$$\mathbf{B}_1 \mathbf{L}_2^T = \begin{bmatrix} 0 & \frac{c_0}{\Delta} \mathbf{M}_1 \\ \frac{c_0}{\Delta} \mathbf{M}_2 & 0 \end{bmatrix} \quad (2.79a)$$

Since the spatial reciprocity is also maintained across the cut, which was used to split up the original grid, the matrix $\mathbf{B}_2 \mathbf{L}_1^T$ must be of the following form:

$$\mathbf{B}_2 \mathbf{L}_1^T = \begin{bmatrix} 0 & -\frac{c_0}{\Delta} \mathbf{M}_2^T \\ -\frac{c_0}{\Delta} \mathbf{M}_1^T & 0 \end{bmatrix} \quad (2.79b)$$

In other words this means: when $\dot{\mathbf{o}}_2$ is related to \mathbf{p}_1 then, $\dot{\mathbf{p}}_1$ is related to \mathbf{o}_2 in a similar fashion. In matrix notation this is made clear by:

$$\mathbf{B}_1 \mathbf{L}_2^T = -(\mathbf{B}_2 \mathbf{L}_1^T)^T \quad (2.80)$$

This allows us now to write the state-space model of the entire grid as:

$$\begin{bmatrix} \mathbf{D}_{\theta,1} & 0 & 0 & 0 \\ 0 & \mathbf{D}_{\theta,2} & 0 & 0 \\ 0 & 0 & \mathbf{D}_{\kappa,1} & 0 \\ 0 & 0 & 0 & \mathbf{D}_{\kappa,2} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}}_1 \\ \dot{\mathbf{o}}_2 \\ \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma,1} & 0 & \frac{c_0}{\Delta} \mathbf{K}_1 & -\frac{c_0}{\Delta} \mathbf{M}_1 \\ 0 & \mathbf{D}_{\sigma,2} & \frac{c_0}{\Delta} \mathbf{M}_2^T & \frac{c_0}{\Delta} \mathbf{K}_2 \\ -\frac{c_0}{\Delta} \mathbf{K}_1^T & -\frac{c_0}{\Delta} \mathbf{M}_2 & \mathbf{D}_{\sigma^*,1} & 0 \\ \frac{c_0}{\Delta} \mathbf{M}_1^T & -\frac{c_0}{\Delta} \mathbf{K}_2^T & 0 & \mathbf{D}_{\sigma^*,2} \end{bmatrix} \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{o}_2 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} \quad (2.81)$$

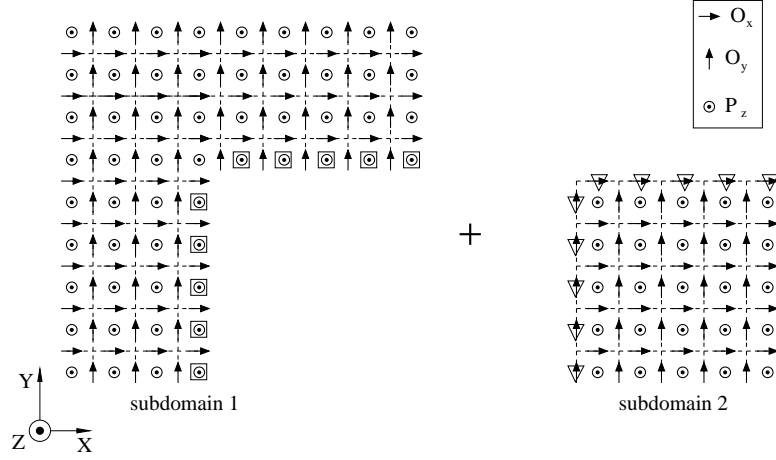


Figure 2.22: A small fraction of a grid that has been cut up into two subdomains. For each subdomain, the input and output variables are of the same type.

In the remainder of the text, when a grid is cut up into several subdomains, the cut will be in such a way that the subdomain boundary field variables are either all of the O -type or all of the P -type, e.g. in Fig. 2.22:

$$\mathbf{u}_1 = \mathbf{y}_2 \subset \mathbf{o}_2 \quad (2.82a)$$

$$\mathbf{u}_2 = \mathbf{y}_1 \subset \mathbf{p}_1 \quad (2.82b)$$

where \subset is used to define a subset. In Fig. 2.22, the output variables of subdomain 1 are P -type field variables, the output variables of subdomain 2 are O -type field variables.

Furthermore, this implies that the input variables of a subdomain are of the opposite type of the input variables of the other subdomain and that either \mathbf{M}_1 or \mathbf{M}_2 is zero.

4.3.2.2 Mathematical interpretation

The previous section described how a grid can be divided into two parts by means of a cut. The different grid parts were then called subdomains. For these subdomains it was possible to write the corresponding mathematical state-space model. In this paragraph this process will be reversed.

Let us start by reconsidering the state-space model of a general uniform orthogonal grid in block matrix form:

$$\begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_\sigma & \frac{c_0}{\Delta} \mathbf{K} \\ -\frac{c_0}{\Delta} \mathbf{K}^T & \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (2.83)$$

This is equivalent to:

$$\begin{cases} \mathbf{D}_\theta \dot{\mathbf{o}} = -\mathbf{D}_\sigma \mathbf{o} - \frac{c_0}{\Delta} \mathbf{K} \mathbf{p} \\ \mathbf{D}_\kappa \dot{\mathbf{p}} = -\mathbf{D}_{\sigma^*} \mathbf{p} + \frac{c_0}{\Delta} \mathbf{K}^T \mathbf{o} \end{cases} \quad (2.84)$$

Both equations are equivalent to a state-space model of the form:

$$\begin{cases} \mathbf{C}_\alpha \dot{\mathbf{x}}_\alpha = -\mathbf{G}_\alpha \mathbf{x}_\alpha + \mathbf{B}_\alpha \mathbf{u}_\alpha \\ \mathbf{y}_\alpha = \mathbf{L}_\alpha^T \mathbf{x}_\alpha \end{cases} \quad \text{for } \alpha = 1, 2 \quad (2.85)$$

where for the first equation:

$$\mathbf{C}_1 = \mathbf{D}_\theta \quad (2.86a)$$

$$\mathbf{G}_1 = \mathbf{D}_\sigma \quad (2.86b)$$

$$\mathbf{B}_1 = -\frac{c_0}{\Delta} \mathbf{K} \quad (2.86c)$$

$$\mathbf{L}_1 = \mathbf{I} \quad (2.86d)$$

$$\mathbf{x}_1 = \mathbf{o} \quad (2.86e)$$

$$\mathbf{u}_1 = \mathbf{p} \quad (2.86f)$$

$$\mathbf{y}_1 = \mathbf{o} \quad (2.86g)$$

and for the second equation:

$$\mathbf{C}_2 = \mathbf{D}_\kappa \quad (2.87a)$$

$$\mathbf{G}_2 = \mathbf{D}_{\sigma^*} \quad (2.87b)$$

$$\mathbf{B}_2 = \frac{c_0}{\Delta} \mathbf{K}^T \quad (2.87c)$$

$$\mathbf{L}_2 = \mathbf{I} \quad (2.87d)$$

$$\mathbf{x}_2 = \mathbf{p} \quad (2.87e)$$

$$\mathbf{u}_2 = \mathbf{o} \quad (2.87f)$$

$$\mathbf{y}_2 = \mathbf{p} \quad (2.87g)$$

By explicitly writing out the block matrix description of the original state-space model (2.84), it has been made clear that splitting up the grid into a subdomain containing all *O*-type fields and a second subdomain with all the *P*-type fields is another choice for dividing a grid.

The importance of this kind of grid splitting will become obvious when the time discretization of the standard FDTD-technique is discussed.

4.3.3 Subgridding

Up to now, only the state-space model associated with a uniform grid was examined. When subgridding is applied, it is still possible to write the

discretized grid as a state-space model, since there is one equation related to each field variable. This equation then always contains:

- the time derivative of that field variable
- this field variable multiplied by a loss factor
- an algebraic sum of neighbouring field variables of the different field type

resulting in a set of first order differential equations. Examining the state-space model in its block matrix notation the following is noted:

$$\begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_\sigma & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (2.88)$$

As was explained before and illustrated by means of Fig. 2.15, the spatial reciprocity property is lost when subgridding is applied. This has its influence on the matrix \mathbf{G} :

$$\mathbf{G}_{12} \neq -\mathbf{G}_{21}^T \quad (2.89)$$

Just as for a uniform orthogonal grid, it is possible to introduce a cut C in the subgridded mesh. Amongst all the cuts possible, some are of particular interest to us. By keeping the following considerations in mind:

- along each side of the cut, it is desirable to have fields of the same type
- try to minimize the number of input and output variables

two interesting cuts will be discussed. The first interesting cut is illustrated in Fig. 2.23. There the cut is made along the transition between the fine grid and the coarse grid. When the fine grid covers n_x coarse cells in the x -direction and n_y coarse cells in the y -direction, then the dimensions for \mathbf{y}_1 (output variables coarse grid) and \mathbf{y}_2 (output variables fine grid) are:

$$\dim[\mathbf{y}_1] = 2(n_x + n_y + 2) \quad (2.90a)$$

$$\dim[\mathbf{y}_2] = 2(n_x + n_y) \quad (2.90b)$$

Thanks to this choice the state-space models associated with both subdomains can be expressed by (2.78). When $\Delta = \Delta_c$, then the non-zero elements of \mathbf{K}_1 take the values -1 and 1 , while the non-zero elements of \mathbf{K}_2 are $-r$ and r . An interesting feature about this cut is that the loss of spatial reciprocity is expressed by:

$$\mathbf{B}_1 \mathbf{L}_2^T \neq (\mathbf{B}_2 \mathbf{L}_1^T)^T \quad (2.91)$$

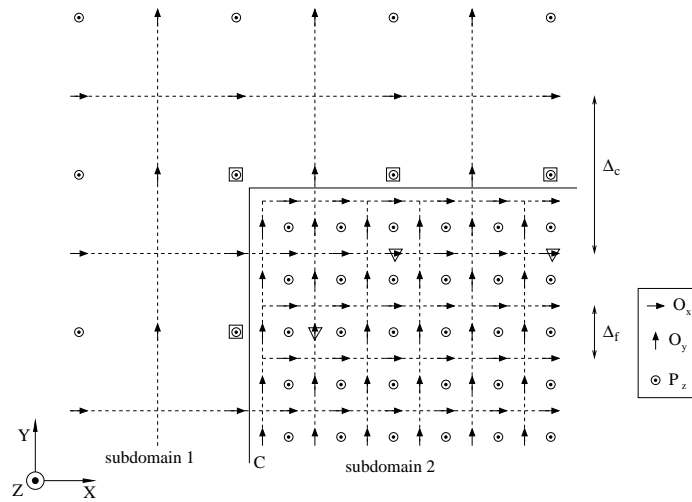


Figure 2.23: A combination of a fine grid and a coarse grid, with $r = 3$, split by a cut C . The output variables of subdomain 1 and subdomain 2 are indicated by \square and ∇ respectively.

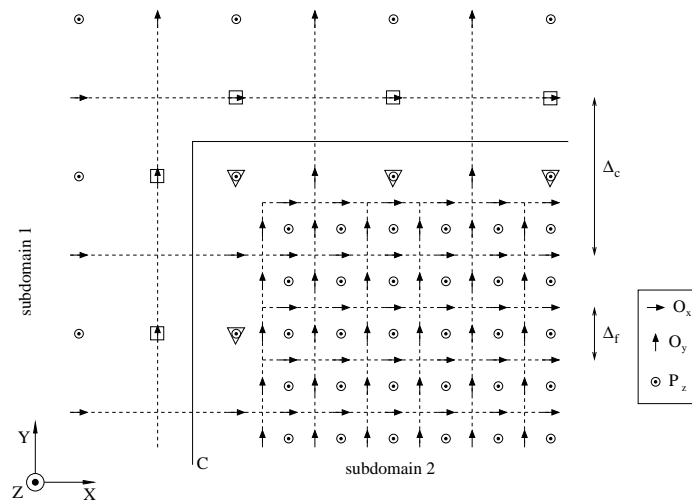


Figure 2.24: A subgridded mesh split up by the cut C . The output variables of subdomain 1 and subdomain 2 are indicated by \square and ∇ respectively.

A second interesting cut is to choose the cut a little bit further away from the fine grid, see Fig. 2.24. Then the dimensions of the output vectors \mathbf{y}_1 and \mathbf{y}_2 are:

$$\dim[\mathbf{y}_1] = 2(n_x + n_y + 4) \quad (2.92a)$$

$$\dim[\mathbf{y}_2] = 2(n_x + n_y + 2) \quad (2.92b)$$

where the interpretation of n_x and n_y has not changed. The dimension of both vectors is now slightly larger than for the previous choice. The lack of spatial reciprocity is now expressed by the matrix \mathbf{G}_2 :

$$\mathbf{G}_{2,12} \neq -\mathbf{G}_{2,21}^T \quad (2.93)$$

BIBLIOGRAPHY

- [1] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 1995.
- [2] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302–307, May 1966.
- [3] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 2000.
- [4] C. J. Railton, I. J. Craddock, and J. B. Schneider, "Improved locally distorted CPFDTD algorithm with provable stability," *Electronics Letters*, vol. 31, no. 18, pp. 1585–1586, Aug. 1995.
- [5] A. Taflove, K. R. Umashankar, B. Beker, F. Harfoush, and K. S. Yee, "Detailed FD-TD analysis of electromagnetic fields penetrating narrow slots and lapped joints in thick conducting screens," *IEEE Trans. Antennas Propagat.*, vol. 36, no. 2, pp. 247–257, Feb. 1988.
- [6] S. S. Zivanovic, K. S. Yee, and K. K. Mei, "A subgridding method for the time-domain finite-difference method to solve Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 39, no. 3, pp. 471–479, Mar. 1991.
- [7] P. Thoma and T. Weiland, "A consistent subgridding scheme for the finite difference time domain method," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 9, no. 5, pp. 359–374, 1996.

- [8] M. J. White, M. F. Iskander, and Z. Huang, "Development of a multigrid FDTD code for three-dimensional applications," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1512–1517, Oct. 1997.
- [9] I. S. Kim and W. J. R. Hoefer, "A local mesh refinement algorithm for the time domain-finite difference method using Maxwell's curl equations," *IEEE Trans. Microwave Theory Tech.*, vol. 38, no. 6, pp. 812–815, June 1990.
- [10] D. T. Prescott and N. V. Shuley, "A method for incorporating different sized cells into the finite-difference time-domain analysis technique," *IEEE Microwave and Guided Wave Letters*, vol. 2, no. 11, pp. 434–436, Nov. 1992.
- [11] M. W. Chevalier, R. J. Luebbers, and V. P. Cable, "FDTD local grid with material traverse," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 3, pp. 411–421, Mar. 1997.
- [12] M. Okoniewski, E. Okoniewska, and M. A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 3, pp. 422–429, Mar. 1997.
- [13] S. Kapoor, "Sub-cellular technique for finite-difference time-domain method," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 5, pp. 673–677, May 1997.
- [14] J. W. Harris and H. Stocker, *Handbook of Mathematics and Computational Science*, Springer-Verlag, 1998.
- [15] K. M. Krishnaiah and C. J. Railton, "Passive equivalent circuit of FDTD: an application to subgridding," *Electronics Letters*, vol. 33, no. 15, pp. 1277–1278, July 1997.
- [16] K. M. Krishnaiah and C. J. Railton, "A stable subgridding algorithm and its application to eigenvalue problems," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 5, pp. 620–628, May 1999.
- [17] W. Yu and R. Mittra, "A new higher-order subgridding method for finite difference time domain (FDTD) algorithm," *IEEE Antennas Propagat. Soc. Int. Symposium, Atlanta GA*, pp. 608–611, 1998.
- [18] W. Yu and R. Mittra, "A new subgridding method for the finite-difference time-domain (FDTD) algorithm," *Microwave Opt. Techn. Letters*, vol. 21, no. 5, pp. 330–333, June 1999.

- [19] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits and Systems*, vol. 22, no. 6, pp. 504–509, June 1975.
- [20] O. L. Elgerd, *Control Systems Theory*, McGraw-Hill Book Company, 1967.
- [21] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time domain electromagnetic field equations," *IEEE Trans. Electromagnetic Compatibility*, vol. 23, no. 4, pp. 377–382, 1981.
- [22] J. P. Bérenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Computational Physics*, vol. 114, no. 1, pp. 185–200, 1994.
- [23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

CHAPTER 3

Finite differences: the temporal problem

1 INTRODUCTION

One of the principal strengths of the FDTD method is the simplicity of the resulting update equations. Simulations containing millions of variables can still easily be computed using a simple desktop computer. Furthermore, the FDTD method lends itself pre-eminently to parallel computing and is thus suited for the simulation of problems containing a very large number of variables. As one of the key aspects of the FDTD grid was the interleaved position of the different field variables, so is the time discretization used in the FDTD method also based on a similar interleaving.

The question on how time can be discretized will be the topic of this chapter. In other words, we will make the step from a problem with a large number of continuous time variables, to a problem described by the same number of variables, only existing at certain discrete, uniformly spaced, points in time. In addition to this, we will try to keep the complexity of the resulting equations as simple as possible and thus trying to limit the number of floating point operations. Three different time discretization schemes will be discussed: the standard FDTD method, the newly introduced subdomain FDTD method and the alternating-direction implicit FDTD (ADI-FDTD) method. All these methods are second order accurate as will be illustrated.

As the acronym clearly indicates, FDTD (finite-difference time-domain) is a time-domain method, which is based on the discretization of the derivatives. However, it not only discretizes the spatial derivatives, as explained in the previous chapter, but it also discretizes the temporal derivatives.

Discretization of the time derivatives is also done using central difference approximations. This then results in the popular leapfrog explicit time-stepping scheme which is the topic of the next section (Section 2).

In Section 3, a more general time-stepping scheme will be presented: the subdomain FDTD method. The main point there is the update equation of a subdomain, which is derived from the state-space description of this subdomain. The resulting time discretized subdomain equations will then be similar to the standard FDTD equations and will be combined with these equations resulting in a more general leapfrog time-stepping scheme.

Finally, in Section 4, a third possible updating scheme will be discussed: the alternating-direction implicit FDTD method. This method has recently been introduced [1] and has the very attractive property of being unconditionally stable.

Whereas subgridding was a focus point in the previous chapter, it will only appear here when the subdomain time discretization approach is discussed. Nevertheless we want to show that the subdomain equations, combined with the standard FDTD equations, result in the innovative algorithm presented in this chapter. The ADI-FDTD method is also part of this chapter since it is interesting, but merely from a theoretical point of view.

As already made clear in Chapter 2, matrix notation can be very convenient when studying FDTD algorithms. It makes a number of properties clearly visible. But since it does involve a more complex, although powerful, mathematical framework and since the matrix notation does not immediately visualize the simplicity of the resulting equations, it is not used in most textbooks [2], [3]. Another reason why textbooks do not employ matrix notation is probably related to the fact that absorbing boundary conditions are used, which interfere with the elegant matrix notation. In this chapter, matrix notation will keep on playing an important role, since it allows for the explanation of the different possibilities for discretizing time in a convenient fashion and since it will help in one of the following chapters, to discuss stability for the different time discretization schemes.

2 STANDARD FDTD METHOD

2.1 Introduction

The temporal discretization used in the standard FDTD method will be discussed here. Some ideas on the approximation of derivatives, more specifically the central difference approximation, will reappear as we go along. We will also elaborate on accuracy and computational complexity of

the standard FDTD method. The FDTD update equations will be derived in two different ways. First, this will be performed using scalar notation, then the same equations will be derived using matrix notation. It is our intention to emphasize matrix notation since it helps to explain the relation between the different time discretization methods.

2.2 Scalar notation

As was explained before, a derivative can be approximated with second order accuracy by means of central difference approximations. For the time derivative of a function $f(t)$, approximated around $t = t_0$, this is:

$$\left. \frac{df(t)}{dt} \right|^{t=t_0} = \frac{f(t_0 + d_t) - f(t_0 - d_t)}{2d_t} + \mathcal{O}(d_t^2) \quad (3.1)$$

Applying this to (2.22a) around the point $t = n\Delta_t$, with $d_t = \Delta_t/2$, we get:

$$\theta_r|_{i+1/2,j} \frac{O_x|_{i+1/2,j}^{n+1/2} - O_x|_{i+1/2,j}^{n-1/2}}{\Delta_t} = -c_0 \frac{P_z|_{i+1/2,j+1/2}^n - P_z|_{i+1/2,j-1/2}^n}{\Delta} - \sigma|_{i+1/2,j} O_x|_{i+1/2,j}^n \quad (3.2)$$

Here Yee's notation, with the x and y coordinate indicated as a subscript and time as a superscript, was used. For example for O_x :

$$O_x|_{i+r,j+s}^{n+q} = O_x(x, y, t) \Big|_{\substack{t=(n+q)\Delta_t \\ x=(i+r)\Delta \\ y=(j+s)\Delta}} \quad (3.3)$$

for $i, j, n \in \mathbb{Z}$ and $0 \leq r, s, q < 1$. Just as the locations for the different field components are spatially interleaved, so are the time instants for O -type and P -type fields almost interleaved in (3.3):

- O_x at half time steps: $t = (n \pm 1/2)\Delta_t$
- P_z at whole time steps: $t = n\Delta_t$

only the last term is not situated at an appropriate time instant. Therefore the value of this last term at $t = n\Delta_t$ will be estimated by an average of the value at two equidistant time instants:

$$O_x|_{i+1/2,j}^n \approx \frac{O_x|_{i+1/2,j}^{n+1/2} + O_x|_{i+1/2,j}^{n-1/2}}{2} \quad (3.4)$$

These two neighbouring time instants were chosen in this way so that they correspond to the time instants of the time derivative approximation. The

accuracy of the approximation in (3.4) can easily be verified by means of Taylor's theorem and is second order accurate:

$$f(t_0) = \frac{f(t_0 + d_t) + f(t_0 - d_t)}{2} + \mathcal{O}(d_t^2) \quad (3.5)$$

since the desired value ($t = n\Delta_t$) is central between the two time instants ($t = (n \pm \frac{1}{2})\Delta_t$) used for the average.

The same can be done for (2.22b) around $t = n\Delta_t$ and for (2.22c) around $t = (n + \frac{1}{2})\Delta_t$. The field variables bringing into account the losses have to be approximated by an average similar as (3.4). Eventually the equations then become:

$$\begin{aligned} \theta_r|_{i+\frac{1}{2},j} \frac{O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - O_x|_{i+\frac{1}{2},j}^{n-\frac{1}{2}}}{\Delta_t} &= -c_0 \frac{P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n - P_z|_{i+\frac{1}{2},j-\frac{1}{2}}^n}{\Delta} \\ &\quad - \sigma|_{i+\frac{1}{2},j} \frac{O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} + O_x|_{i+\frac{1}{2},j}^{n-\frac{1}{2}}}{2} \end{aligned} \quad (3.6a)$$

$$\begin{aligned} \theta_r|_{i,j+\frac{1}{2}} \frac{O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - O_y|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}}}{\Delta_t} &= c_0 \frac{P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n - P_z|_{i-\frac{1}{2},j+\frac{1}{2}}^n}{\Delta} \\ &\quad - \sigma|_{i,j+\frac{1}{2}} \frac{O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} + O_y|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}}}{2} \end{aligned} \quad (3.6b)$$

$$\begin{aligned} \kappa_r|_{i+\frac{1}{2},j+\frac{1}{2}} \frac{P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n}{\Delta_t} &= c_0 \left[\frac{O_y|_{i+1,j+\frac{1}{2}}^{n+\frac{1}{2}} - O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta} \right. \\ &\quad \left. - \frac{O_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} - O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}}}{\Delta} \right] - \sigma^*|_{i+\frac{1}{2},j+\frac{1}{2}} \frac{P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} + P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n}{2} \end{aligned} \quad (3.6c)$$

In each of these equations the following is true:

- all the O -type fields appear only at half time steps: $t = (n + \frac{1}{2})\Delta_t$ where $n \in \mathbb{N}$
- all the P -type fields appear only at whole time steps: $t = n\Delta_t$ where $n \in \mathbb{N}$

The equations (3.6) can now be rewritten as:

$$\begin{aligned} O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} &= \frac{\theta_r|_{i+\frac{1}{2},j} - \frac{\Delta_t \sigma|_{i+\frac{1}{2},j}}{2}}{\theta_r|_{i+\frac{1}{2},j} + \frac{\Delta_t \sigma|_{i+\frac{1}{2},j}}{2}} O_x|_{i+\frac{1}{2},j}^{n-\frac{1}{2}} \\ &\quad - \frac{k}{\theta_r|_{i+\frac{1}{2},j} + \frac{\Delta_t \sigma|_{i+\frac{1}{2},j}}{2}} (P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n - P_z|_{i+\frac{1}{2},j-\frac{1}{2}}^n) \end{aligned} \quad (3.7a)$$

$$O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{\theta_r|_{i,j+\frac{1}{2}} - \frac{\Delta_t \sigma|_{i,j+\frac{1}{2}}}{2}}{\theta_r|_{i,j+\frac{1}{2}} + \frac{\Delta_t \sigma|_{i,j+\frac{1}{2}}}{2}} O_y|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{k}{\theta_r|_{i,j+\frac{1}{2}} + \frac{\Delta_t \sigma|_{i,j+\frac{1}{2}}}{2}} \left(P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n - P_z|_{i-\frac{1}{2},j+\frac{1}{2}}^n \right) \quad (3.7b)$$

$$P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \frac{\kappa_r|_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{\Delta_t \sigma^*|_{i+\frac{1}{2},j+\frac{1}{2}}}{2}}{\kappa_r|_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{\Delta_t \sigma^*|_{i+\frac{1}{2},j+\frac{1}{2}}}{2}} P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n + \frac{k}{\kappa_r|_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{\Delta_t \sigma^*|_{i+\frac{1}{2},j+\frac{1}{2}}}{2}} \left(O_y|_{i+1,j+\frac{1}{2}}^{n+\frac{1}{2}} - O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - O_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} + O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right) \quad (3.7c)$$

where

$$k = \frac{c_0 \Delta t}{\Delta} \quad (3.8)$$

These equations all have a similar structure with respect to the time instants used and to the fields used in the equations. This structure can be summarized as:

- at the left hand side: the updated field variable at a new time step
- at the right hand side: the field variable at the previous time step
- and the neighbouring fields at an intermediate time instant

Each equation in (3.7) enables the calculation of a field variable at a new time instant just by knowing the value at the previous time instant and the values of the neighbouring field variables at an intermediate time instant. It is noted that all the coefficients in each equation are time independent.

The equations (3.7) can now be used in an iterative way to simulate the time behaviour of the fields. Once the coefficients, which are time-independent have been calculated, the algorithm proceeds as follows:

1. set $n = 0$ and start with fields equal to zero
2. use (3.7a) and (3.7b) to advance the values of the O_x and O_y fields one time step and add possible source values
3. use (3.7c) to advance the values of the P_z fields one time step and add possible source values
4. while $n \leq n_{\text{final}}$ increment n and go back to step 2

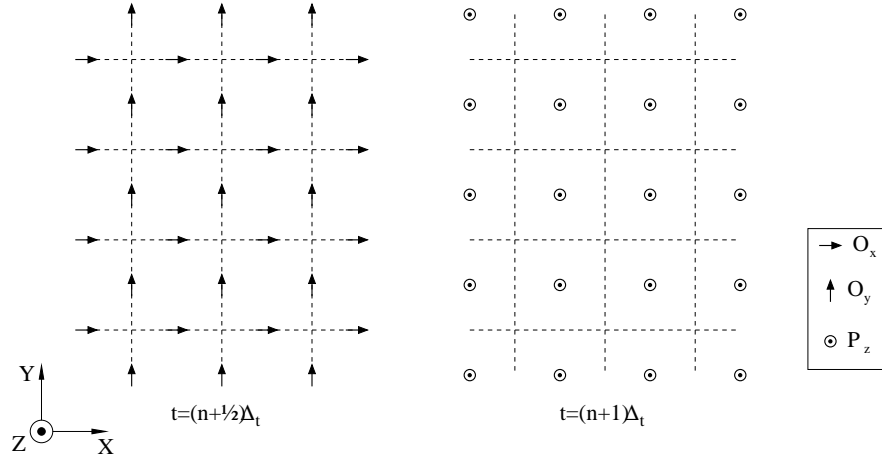


Figure 3.1: Small part of a regular uniform FDTD grid indicating which field variables are updated at whole time instants and which field variables are updated at half time instants.

In Fig. 3.1 a small part of a regular uniform grid is shown twice. The left panel shows the field variables updated at $t = (n + 1/2)\Delta_t$ and the right panel indicates which field variables are updated at $t = (n + 1)\Delta_t$. This kind of algorithm is often referred to as a *leapfrog* time stepping scheme, since not all fields are updated at the same time instant. The different fields advance in time in an alternating way: the *O*-type fields are updated after the *P*-type fields, and the *P*-type fields are updated after the *O*-type fields. Also remark that the equations are *explicit*: the simulation involves no matrix computations.

Of course some kind of source needs to be added. Two of the most important sources are: the plane wave source [4] and the current sources [5]. More details concerning sources can be found in [2] and references therein.

The implementation of (3.7) is numerically very interesting since all the normalizations introduced in Section 2.2 have, depending on the materials present in the simulation domain, resulted in small coefficients. The coefficients

$$\frac{\theta_r - \frac{\Delta_t \sigma}{2}}{\theta_r + \frac{\Delta_t \sigma}{2}} \quad \text{and} \quad \frac{\kappa_r - \frac{\Delta_t \sigma^*}{2}}{\kappa_r + \frac{\Delta_t \sigma^*}{2}} \quad (3.9)$$

become closer to one as the losses become smaller. The parameter k is

typically a bit smaller than $1/\sqrt{2}$, resulting in values of

$$\frac{k}{\theta_r + \frac{\Delta_t \sigma}{2}} \quad \text{and} \quad \frac{k}{\kappa_r + \frac{\Delta_t \sigma^*}{2}} \quad (3.10)$$

in magnitude smaller than one, depending on the relative constitutive parameters θ_r and κ_r . It also means that the O_x , O_y and P_z fields are similar in magnitude, so no multiplications of very large and very small numbers are involved.

For lossless media, i.e. $\sigma = 0$ and $\sigma^* = 0$, the equations (3.7) are simplified to:

$$O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = O_x|_{i+\frac{1}{2},j}^{n-\frac{1}{2}} - \frac{k}{\theta_r|_{i+\frac{1}{2},j}} \left(P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n - P_z|_{i+\frac{1}{2},j-\frac{1}{2}}^n \right) \quad (3.11a)$$

$$O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = O_y|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{k}{\theta_r|_{i,j+\frac{1}{2}}} \left(P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n - P_z|_{i-\frac{1}{2},j+\frac{1}{2}}^n \right) \quad (3.11b)$$

$$P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = P_z|_{i+\frac{1}{2},j+\frac{1}{2}}^n + \frac{k}{\kappa_r|_{i+\frac{1}{2},j+\frac{1}{2}}} \left(O_y|_{i+1,j+\frac{1}{2}}^{n+\frac{1}{2}} - O_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - O_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} + O_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right) \quad (3.11c)$$

Just as was the case for the spatial derivatives, interleaving the time instants for the various field types results in higher accuracy. When for each equation the time derivative would be approximated around $t = n\Delta_t$ by, e.g. for O_x :

$$\left. \frac{dO_x}{dt} \right|_n \approx \frac{O_x|^{n+1} - O_x|^{n-1}}{2\Delta_t} \quad (3.12)$$

all field variables would be updated at whole time steps. Yet, although the scheme would be second order accurate, $\mathcal{O}(\Delta_t^2)$, this error would be four times as large. Furthermore it would require the storage of two field values per field variable: the values at two previous time steps, whereas for the standard FDTD method only one value needs to be stored.

2.3 Matrix notation

It is possible to derive the same equations, using the same reasoning but starting from the state-space model representing the uniform orthogonal grid. We supposed the grid to be terminated by zero fields:

$$\begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_\sigma & \frac{c_0}{\Delta} \mathbf{K} \\ -\frac{c_0}{\Delta} \mathbf{K}^T & \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (3.13)$$

However, time discretization will now be applied to vectors instead of single variables. This will lead to the same equations, yet this time in matrix notation.

By explicitly writing out this state-space model using the different blocks, we obtain:

$$\mathbf{D}_\theta \dot{\mathbf{o}} = -\frac{c_0}{\Delta} \mathbf{K} \mathbf{p} - \mathbf{D}_\sigma \mathbf{o} \quad (3.14a)$$

$$\mathbf{D}_\kappa \dot{\mathbf{p}} = \frac{c_0}{\Delta} \mathbf{K}^T \mathbf{o} - \mathbf{D}_{\sigma^*} \mathbf{p} \quad (3.14b)$$

Using for (3.14a), around $t = n\Delta_t$, central difference approximations to approximate $\dot{\mathbf{o}}$ and averaging to approximate \mathbf{o} :

$$\dot{\mathbf{o}}|n \approx \frac{1}{\Delta_t} (\mathbf{o}|^{n+1/2} - \mathbf{o}|^{n-1/2}) \quad (3.15a)$$

$$\mathbf{o}|n \approx \frac{1}{2} (\mathbf{o}|^{n+1/2} + \mathbf{o}|^{n-1/2}) \quad (3.15b)$$

and, using for (3.14b), around $t = (n + 1/2)\Delta_t$, central difference approximations for $\dot{\mathbf{p}}$ and averaging for \mathbf{p} :

$$\dot{\mathbf{p}}|^{n+1/2} \approx \frac{1}{\Delta_t} (\mathbf{p}|^{n+1} - \mathbf{p}|^n) \quad (3.16a)$$

$$\mathbf{p}|^{n+1/2} \approx \frac{1}{2} (\mathbf{p}|^{n+1} + \mathbf{p}|^n) \quad (3.16b)$$

allows us to approximate equations (3.14) as:

$$\frac{1}{\Delta_t} \mathbf{D}_\theta (\mathbf{o}|^{n+1/2} - \mathbf{o}|^{n-1/2}) = -\frac{c_0}{\Delta} \mathbf{K} \mathbf{p}|^n - \frac{1}{2} \mathbf{D}_\sigma (\mathbf{o}|^{n+1/2} + \mathbf{o}|^{n-1/2}) \quad (3.17a)$$

$$\frac{1}{\Delta_t} \mathbf{D}_\kappa (\mathbf{p}|^{n+1} - \mathbf{p}|^n) = \frac{c_0}{\Delta} \mathbf{K}^T \mathbf{o}|^{n+1/2} - \frac{1}{2} \mathbf{D}_{\sigma^*} (\mathbf{p}|^{n+1} + \mathbf{p}|^n) \quad (3.17b)$$

These equations can be rewritten as:

$$\begin{aligned} \mathbf{o}|^{n+1/2} &= \left(\mathbf{D}_\theta + \frac{\Delta_t}{2} \mathbf{D}_\sigma \right)^{-1} \left(\mathbf{D}_\theta - \frac{\Delta_t}{2} \mathbf{D}_\sigma \right) \mathbf{o}|^{n-1/2} \\ &\quad - k \left(\mathbf{D}_\theta + \frac{\Delta_t}{2} \mathbf{D}_\sigma \right)^{-1} \mathbf{K} \mathbf{p}|^n \end{aligned} \quad (3.18a)$$

$$\begin{aligned} \mathbf{p}|^{n+1} &= \left(\mathbf{D}_\kappa + \frac{\Delta_t}{2} \mathbf{D}_{\sigma^*} \right)^{-1} \left(\mathbf{D}_\kappa - \frac{\Delta_t}{2} \mathbf{D}_{\sigma^*} \right) \mathbf{p}|^n \\ &\quad + k \left(\mathbf{D}_\kappa + \frac{\Delta_t}{2} \mathbf{D}_{\sigma^*} \right)^{-1} \mathbf{K}^T \mathbf{o}|^{n+1/2} \end{aligned} \quad (3.18b)$$

These equations are nothing but the standard FDTD equations (3.7), for all the field variables present in the grid. The matrices

$$\left(\mathbf{D}_\theta + \frac{\Delta_t}{2}\mathbf{D}_\sigma\right)^{-1} \left(\mathbf{D}_\theta - \frac{\Delta_t}{2}\mathbf{D}_\sigma\right) \quad (3.19a)$$

and

$$\left(\mathbf{D}_\kappa + \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*}\right)^{-1} \left(\mathbf{D}_\kappa - \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*}\right) \quad (3.19b)$$

might look very impressive but, since all \mathbf{D} matrices are diagonal, they are diagonal themselves. The elements on the diagonal are the coefficients of (3.9). Similarly the matrices

$$k \left(\mathbf{D}_\theta + \frac{\Delta_t}{2}\mathbf{D}_\sigma\right)^{-1} \quad (3.20a)$$

and

$$k \left(\mathbf{D}_\kappa + \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*}\right)^{-1} \quad (3.20b)$$

are diagonal matrices where the coefficients (3.10), are the diagonal elements.

The update equations (3.18) can be written as one system:

$$\begin{aligned} & \begin{bmatrix} \mathbf{I} & 0 \\ -k \left(\mathbf{D}_\kappa + \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*}\right)^{-1} \mathbf{K}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n+1/2} \\ \mathbf{p}|^{n+1} \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathbf{D}_\theta + \frac{\Delta_t}{2}\mathbf{D}_\sigma\right)^{-1} \left(\mathbf{D}_\theta - \frac{\Delta_t}{2}\mathbf{D}_\sigma\right) & -k \left(\mathbf{D}_\theta + \frac{\Delta_t}{2}\mathbf{D}_\sigma\right)^{-1} \mathbf{K} \\ 0 & \left(\mathbf{D}_\kappa + \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*}\right)^{-1} \left(\mathbf{D}_\kappa - \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*}\right) \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n-1/2} \\ \mathbf{p}|^n \end{bmatrix} \end{aligned} \quad (3.21)$$

Implementing this is computationally as performant: the right hand side is a matrix vector product involving a sparse matrix. The matrix at the left hand side is sparse and lower triangular, meaning that the system can be solved efficiently by means of back-substitution.

For lossless media this all simplifies to:

$$\mathbf{o}|^{n+1/2} = \mathbf{o}|^{n-1/2} - k \mathbf{D}_\theta^{-1} \mathbf{K} \mathbf{p}|^n \quad (3.22a)$$

$$\mathbf{p}|^{n+1} = \mathbf{p}|^n + k \mathbf{D}_\kappa^{-1} \mathbf{K}^T \mathbf{o}|^{n+1/2} \quad (3.22b)$$

or

$$\begin{bmatrix} \mathbf{I} & 0 \\ -k \mathbf{D}_\kappa^{-1} \mathbf{K}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n+1/2} \\ \mathbf{p}|^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -k \mathbf{D}_\theta^{-1} \mathbf{K} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n-1/2} \\ \mathbf{p}|^n \end{bmatrix} \quad (3.23)$$

2.4 Computational considerations

As far as memory requirements are concerned, the standard FDTD method only involves the storage of one value per field variable. This value is then overwritten at each time step. Apart from this, two coefficients, (3.9) and (3.10), per field variable need to be stored. When lossless media are involved this is only one coefficient. In the worst case, this sums up to $3N$ floating point numbers that need to be stored, when there are N field variables present inside the grid. We assumed here that material properties can change from grid point to grid point.

When considering the computational complexity, a straightforward implementation of the equations (3.7) requires at least 1 and at most two multiplications per time step. Each update also demands 2 additions for O -type fields and 4 for P -type fields. We will use *flops*, floating point operations, to quantify the complexity of an algorithm. For a standard FDTD simulation then, by taking into account the relative occurrence of the different types of fields, at most $14N/3$ flops per time step are needed.

Clearly for the standard FDTD method, the memory requirements and the number of floating point operations per time step is linear to the number of field variables used. This is an important property since it is the main reason why the FDTD method can be used even when a very large number of variables is required.

3 SUBDOMAIN FDTD METHOD

3.1 Subdomain types

For the subdomain FDTD method we start from a grid that has been divided into subdomains by means of a cut C . As was explained at the end of Section 4.3.2.1 of Chapter 2 we suppose each subdomain to have boundary fields of the same type. In this way a subdomain can be characterized as one of two possible types:

1. a *O-type subdomain*: all the field variables at the boundary of the subdomain are either O_x or O_y
2. a *P-type subdomain*: all the field variables at the subdomain boundary are P_z

Introducing several cuts, where the various cuts do not intersect, a subdomain can be associated to each cut. Then, the subdomain is the part of

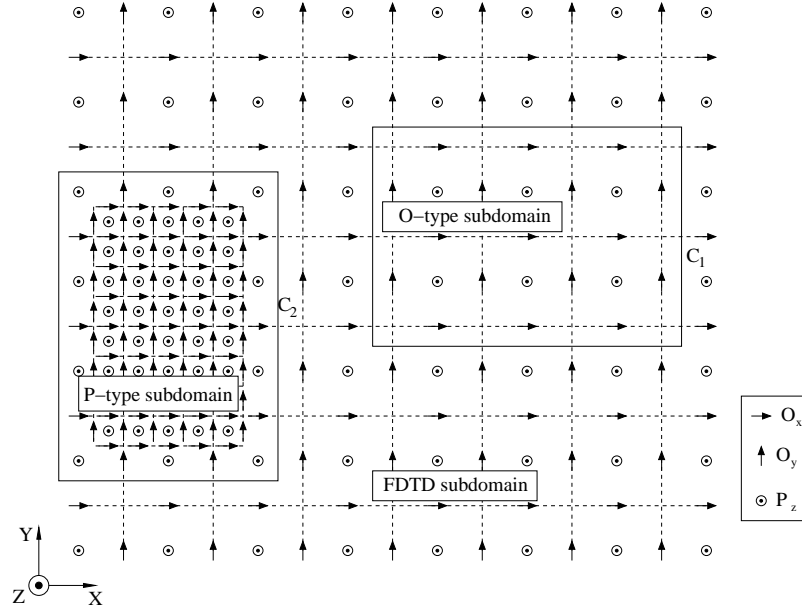


Figure 3.2: Cut C_1 has cut out an O -type regular subdomain, cut C_2 has cut out a P -type subgridded subdomain and the remaining grid is the FDTD subdomain.

the grid inside the cut. Each of these subdomains is either O -type or P -type. The remaining part of the initial grid is clearly also a subdomain, but it is, in general, not of one specific type. This remainder will be called the *FDTD subdomain*. In Fig. 3.2 a sample of a grid with two cuts is shown. Cut C_1 determines a O -type subdomain and cut C_2 determines a P -type subdomain. The remainder of the grid is called the FDTD subdomain. A finer discretization can be used inside the O -type and the P -type subdomains. This is illustrated for the subdomain determined by C_2 .

3.2 Subdomain time discretization

Time discretization of these subdomains can now be performed, corresponding to their type, in a way similar to the time discretization of the normal FDTD field variables. For a O -type subdomain, written in state-space description as:

$$\begin{cases} \mathbf{C}_1 \dot{\mathbf{x}}_1 = -\mathbf{G}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u}_1 \\ \mathbf{y}_1 = \mathbf{L}_1^T \mathbf{x}_1 \end{cases} \quad (3.24)$$

based on approximations for $\dot{\mathbf{x}}_1$ and \mathbf{x}_1 , similar as in (3.15), an approximation around $t = n\Delta_t$ leads for the first equation to:

$$\mathbf{C}_1 \frac{1}{\Delta_t} (\mathbf{x}_1|^{n+1/2} - \mathbf{x}_1|^{n-1/2}) = -\mathbf{G}_1 \frac{1}{2} (\mathbf{x}_1|^{n+1/2} + \mathbf{x}_1|^{n-1/2}) + \mathbf{B}\mathbf{u}_1|^{n-1/2} \quad (3.25a)$$

The second equation that selects the field variables used as outputs (expressed by matrix \mathbf{L}_1) simply becomes:

$$\mathbf{y}_1|^{n+1/2} = \mathbf{L}_1^T \mathbf{x}_1|^{n+1/2} \quad (3.25b)$$

These equations can now be rewritten as:

$$\begin{cases} \mathbf{x}_1|^{n+1/2} = \left(\mathbf{C}_1 + \frac{\Delta_t}{2} \mathbf{G}_1 \right)^{-1} \left(\mathbf{C}_1 - \frac{\Delta_t}{2} \mathbf{G}_1 \right) \mathbf{x}_1|^{n-1/2} \\ \quad + \Delta_t \left(\mathbf{C}_1 + \frac{\Delta_t}{2} \mathbf{G}_1 \right)^{-1} \mathbf{B}_1 \mathbf{u}_1|^{n-1/2} \\ \mathbf{y}_1|^{n+1/2} = \mathbf{L}_1^T \mathbf{x}_1|^{n+1/2} \end{cases} \quad (3.26)$$

Since, for this subdomain, the field variables of \mathbf{y}_1 are either O_x or O_y and the field variables of \mathbf{u}_1 are P_z , the nature of the field variables of the first equation of (3.26) can be summarized as:

- at the left hand side: the field variables at the new time step $t = (n + 1/2)\Delta_t$
- at the right hand side: the field variables at the previous time step $t = (n - 1/2)\Delta_t$ and
- the values of the neighbouring field variables, at an intermediate time instant $t = n\Delta_t$

This is very similar to the standard FDTD update equations (3.7). In addition there is a second equation in (3.26), extracting the output variables of the subdomain. Bringing this together we can see that the P_z fields of \mathbf{u}_1 are needed at $t = n\Delta_t$ and that the update equation (3.26) gives the O -type fields at $t = (n + 1/2)\Delta_t$. Since the subdomain only communicates with the surrounding grid via \mathbf{u}_1 and \mathbf{y}_1 , it is possible to use this update equation in combination with the standard FDTD equations.

For a P -type subdomain a similar reasoning can be applied. The state-space description of a P -type subdomain looks like:

$$\begin{cases} \mathbf{C}_2 \dot{\mathbf{x}}_2 = -\mathbf{G}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u}_2 \\ \mathbf{y}_2 = \mathbf{L}_2^T \mathbf{x}_2 \end{cases} \quad (3.27)$$

and it can be time discretized by using approximations for $\dot{\mathbf{x}}_2$ and \mathbf{x}_2 , around $t = (n + 1/2)\Delta_t$, similar as in (3.17). This then results in:

$$\begin{cases} \mathbf{x}_2|^{n+1} = \left(\mathbf{C}_2 + \frac{\Delta_t}{2}\mathbf{G}_2\right)^{-1} \left(\mathbf{C}_2 - \frac{\Delta_t}{2}\mathbf{G}_2\right) \mathbf{x}_2|^n \\ \quad + \Delta_t \left(\mathbf{C}_2 + \frac{\Delta_t}{2}\mathbf{G}_2\right)^{-1} \mathbf{B}_2 \mathbf{u}_2|^{n+1/2} \\ \mathbf{y}_2|^{n+1} = \mathbf{L}_2^T \mathbf{x}_2|^{n+1} \end{cases} \quad (3.28)$$

Equivalently as for the *O*-type subdomain, the *O*-type fields in \mathbf{u}_2 are required at $t = (n + 1/2)\Delta_t$ and (3.28) gives the *P*-type fields at $t = (n + 1)\Delta_t$, making it possible to use this update equation in combination with the standard FDTD equations. As a consequence, the time discretization of the field variables in the FDTD subdomain is performed as in the standard FDTD method (3.7).

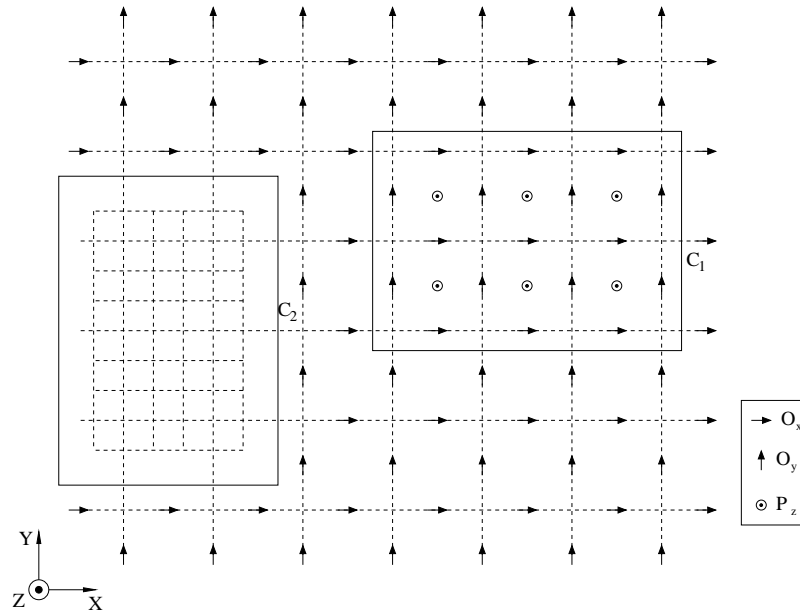
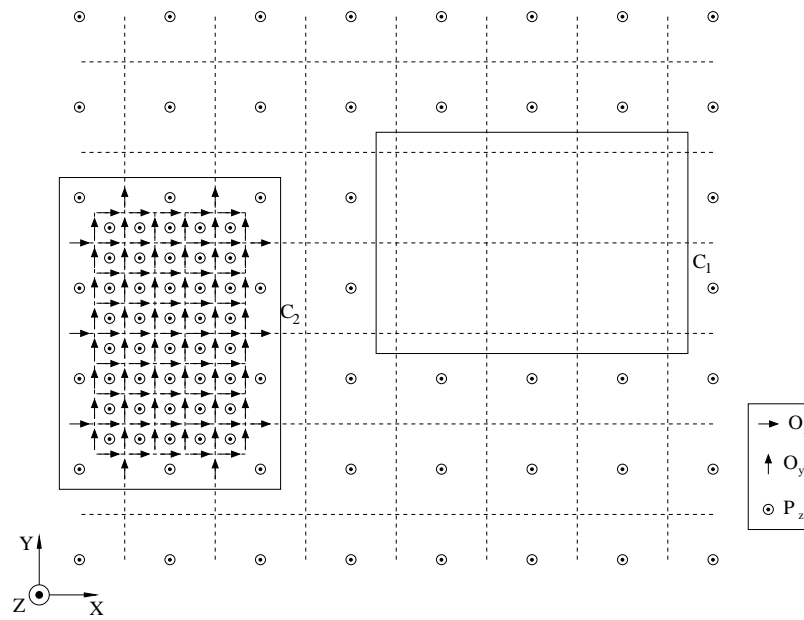
Since the fields and the subdomains present in the simulation domain can step in time hand in hand, a new iterative and more general algorithm can be presented:

1. set $n = 0$ and start with field variables equal to zero
2. use (3.26) to advance the *O*-type subdomains and use (3.7a) and (3.7b) to advance the O_x and O_y field variables in the FDTD subdomain one time step ; add possible source values
3. use (3.28) to advance the *P*-type subdomains and use (3.7c) to advance the P_z field variables in the FDTD subdomain one time step; add possible source values
4. while $n \leq n_{\text{final}}$: increment n and go back to step 2

This algorithm is called the *subdomain FDTD* algorithm. In Fig. 3.3, for the grid illustrated in Fig. 3.2, the field variables that are updated at $t = (n + 1/2)\Delta_t$ are shown, in Fig. 3.4 the field variables updated at $t = (n + 1)\Delta_t$ are shown.

We note also that a single O_x or O_y variable can be seen as a mini *O*-type subdomain and similar for a single P_z variable.

Earlier, in Section 4.3.2.2 of Chapter 2, it was explained that a FDTD subdomain is mathematically equivalent to two smaller subdomains. The first subdomain contains all the O_x and O_y fields and is then clearly a *O*-type subdomain. The second subdomain contains all the P_z fields of the FDTD subdomain and is then a *P*-type subdomain. Based on this interpretation time can then be discretized around $t = n\Delta_t$ for all the *O*-type subdomains and around $t = (n + 1/2)\Delta_t$ for all the *P*-type subdomains.

Figure 3.3: The fields updated at $t = (n + 1/2)\Delta_t$.Figure 3.4: The fields updated at $t = (n + 1)\Delta_t$.

3.3 Computational considerations

Implementing the subdomain equations (3.26) and (3.28), is expensive as far as computational requirements and memory is concerned. Both memory requirements and computational cost (in flops per time step) rise quadratically since the matrix:

$$\left(\mathbf{C} + \frac{\Delta_t}{2}\mathbf{G}\right)^{-1} \left(\mathbf{C} - \frac{\Delta_t}{2}\mathbf{G}\right) \quad (3.29)$$

is in general a full matrix. Remember: the inverse of a sparse matrix is not necessarily sparse. For the same reason the matrix

$$\left(\mathbf{C} + \frac{\Delta_t}{2}\mathbf{G}\right)^{-1} \mathbf{B} \quad (3.30)$$

also needs to be stored as a full matrix. The dimensions of these matrices depend on the number of field variables inside the subdomain, and for (3.30) also on the dimension of \mathbf{u} .

The standard FDTD equations rise linearly, although the FDTD subdomain is a combination of two subdomains, the *O*-type subdomain containing all *O*-type field variables and the *P*-type subdomain containing all *P*-type field variables. The subdomain update equations for these 2 subdomains, (3.26) and (3.28), do not rise quadratically since the \mathbf{C} and \mathbf{G} matrices involved (2.86-2.87) and their inverse are diagonal, thus maintaining sparsity.

A possibility to minimize the iteration cost is to compute a real block diagonal form of (3.29):

$$\left(\mathbf{C} + \frac{\Delta_t}{2}\mathbf{G}\right)^{-1} \left(\mathbf{C} - \frac{\Delta_t}{2}\mathbf{G}\right) = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^{-1} \quad (3.31)$$

where $\mathbf{\Lambda}$ has on its diagonal either the real eigenvalues or for the complex eigenvalues, which come in pairs, $a \pm ib$, a 2×2 block of the form

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \quad (3.32)$$

on the diagonal. This diagonalization calculates the similarity transformation that transforms the real iteration matrix (3.29), in a real sparse matrix containing lowest number of non-zero elements, $\mathbf{\Lambda}$. The matrix \mathbf{P} is also real. In this way, the subdomain update equation, e.g. (3.26), becomes:

$$\begin{cases} \mathbf{P}^{-1}\mathbf{x}_1|^{n+1/2} = \mathbf{\Lambda}\mathbf{P}^{-1}\mathbf{x}_1|^{n-1/2} + \Delta_t\mathbf{P}^{-1} \left(\mathbf{C}_1 + \frac{\Delta_t}{2}\mathbf{G}_1\right)^{-1} \mathbf{B}_1\mathbf{u}_1|^n \\ \mathbf{y}_1|^{n+1/2} = \mathbf{L}_1^T \mathbf{P} \mathbf{P}^{-1} \mathbf{x}_1|^{n+1/2} \end{cases} \quad (3.33)$$

or introducing the new variable $\mathbf{x}' = \mathbf{P}^{-1}\mathbf{x}$

$$\begin{cases} \mathbf{x}'_1|^{n+1/2} = \Lambda \mathbf{x}'_1|^{n-1/2} + \Delta_t \mathbf{P}^{-1} \left(\mathbf{C}_1 + \frac{\Delta_t}{2} \mathbf{G}_1 \right)^{-1} \mathbf{B}_1 \mathbf{u}_1|^{n-1/2} \\ \mathbf{y}_1|^{n+1/2} = \mathbf{L}_1^T \mathbf{P} \mathbf{x}'_1|^{n+1/2} \end{cases} \quad (3.34)$$

This equation is linear, but becomes quickly too expensive to compute for the subdomains that will be used.

An alternative approach would be to solve

$$\left(\mathbf{C} + \frac{\Delta_t}{2} \mathbf{G} \right) \mathbf{x}|^{n+1} = \left(\mathbf{C} - \frac{\Delta_t}{2} \mathbf{G} \right) \mathbf{x}|^n + \Delta_t \mathbf{B} \mathbf{u}|^{n+1/2} \quad (3.35)$$

for every time step by means of sparse matrix computations. This would then be an *implicit* algorithm, requiring not as much memory.

We will not try to answer the question which approach is best, but we will avoid this question by using a reduced order modeling (ROM) technique. This ROM technique generates an approximated state-space model which has a smaller set of variables. This will be the topic of the next chapter. The smaller set of variables will result in smaller matrices (3.29) and (3.30). Furthermore, the real block diagonalization of (3.31) will, once the ROM has been performed, be very easy to compute and will accelerate the algorithm even further.

It also needs to be stated that in this way, subgridding can be elegantly combined with the normal FDTD equations, by incorporating it inside a subdomain. The use of subgridding, up to now, involved the either the use of a small time step or the use of two different time steps: a large time step for the coarse grid and a small time step for the fine grid. The ratio of the coarse time step to the fine time step is then equal to the refinement ratio r , e.g. [6], [7]. The different pace in time stepping is then often based on time extrapolation [6] or travelling wave equations [8]. Thanks to the use of the standard FDTD equations in combination with the subdomain update equations (3.26) and (3.28), which can be applied as well to a normal grid as to a partly subgridded mesh, time stepping is now easier and more elegant since only one time step is involved. The implications this has on the size of the time step will be discussed in Chapter 5.

3.4 The subdomain FDTD method generalized

We have, up to now, investigated grids which consist of subdomains, of two possible types, the *O*-type and the *P*-type, linked together through one surrounding grid, that was time discretized in the standard FDTD fashion. This way of discretizing time is a more specific case of what will be explained in this paragraph.

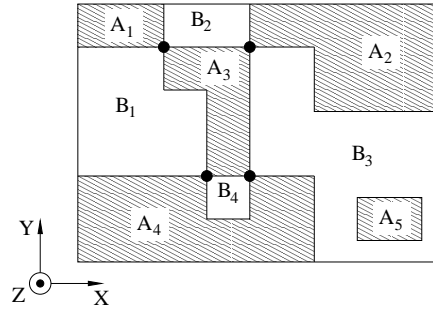


Figure 3.5: A problem space divided in subdomains, where each subdomain is either of type A or of type B. Each subdomain is only surrounded by subdomains of the other type.

Divide the problem space in a number of subdomains. There is no restriction on the field variables at the boundary of each subdomain. These boundary field variables, belonging to one subdomain, can be both *O*- and *P*-type. However, there is one extra restriction, we want to be able to distinguish each one of these subdomains to be of one out of two possible types, say A and B, and this so that each subdomain has neighbouring subdomains of the other type.

In Fig. 3.5, a problem space has been divided in 9 subdomains: 5 subdomains of type A (A_1, A_2, \dots, A_5) and 4 subdomains of type B (B_1, B_2, B_3, B_4). It is clear that for a subdomain located inside another subdomain, e.g. A_5 is located inside B_3 , our extra condition is fulfilled. For the other subdomains it can be verified that A-type subdomains are only surrounded by B-type subdomains and vice versa. Take for example A_3 , this subdomain is surrounded by B_1, B_2, B_3 and B_4 , which are all subdomains of the B-type.

This restriction implies that, in a point inside the problem space, it is not possible that 3 subdomains come together. A point where four subdomains come together does exist (see • in Fig. 3.5).

A more intuitive explanation is to colour code each subdomain, say black for the A-type subdomains and white for the B-type subdomains, see Fig. 3.5. Then each boundary between two subdomains must be a boundary between a black subdomain and a white subdomain. If each subdomain were a square, this would then result in a chessboard kind of grid.

It is now easy to introduce the leapfrog time stepping scheme, where the A-type subdomains are updated at $t = (n + 1/2)\Delta_t$ and the B-type subdomains at $t = (n + 1)\Delta_t$:

1. set $n=0$ and start with field variables equal to zero

2. use (3.26) to advance the values of the A -type subdomains one time step
3. use (3.28) to advance the values of the B -type subdomains one time step
4. while $n \leq n_{\text{final}}$: increment n and go back to 2

The input variables of a type B subdomain are needed at $t = (n + \frac{1}{2})\Delta_t$, see (3.28). These field variables are the output variables of the surrounding subdomains. Since the surrounding subdomains are of the other type, type A , and updated at $t = (n + \frac{1}{2})\Delta_t$, these input variables are available at the appropriate moment in time. The same is valid for the input variables of the type A subdomain. This explains why it was necessary to introduce two kinds of subdomains. At half time steps the A -type or black subdomains are updated, at whole time steps the B -type or white subdomains are updated. The sources have been added as extra input variables in \mathbf{u} in (3.26) and in (3.28) and need to be calculated at the appropriate time instants. This time stepping scheme is second order accurate. In [9] this kind of time stepping was introduced and applied to a waveguide problem.

Although no restriction was imposed on boundary field variables of each subdomain, it is possible to go one step further when the boundary field variables of a certain subdomain are of one specific type: either O or P . For this specific subdomain, it is possible to discretize time in the normal FDTD fashion instead of (3.26) or (3.28). The time instant, however, at which the field variables are updated now depends on the type of the subdomain (A or B) and on the type of the boundary field variables (O or P):

- the field variables belonging to the subdomain and of the type of the boundary field variables of that subdomain are updated together with subdomains of the same subdomain type
- the field variables of the other type are updated together with the subdomains of the other subdomain type

The time instants at which the field variables, belonging to a subdomain with only one type of field variables at the boundary, need to be updated are illustrated more clearly in Table 3.1.

Table 3.1: Time instants at which field variables are updated.

| type of subdomain | type of boundary field variables | O_x, O_y | P_z |
|-------------------|----------------------------------|---------------------|---------------------|
| A | O_x, O_y | $(n + 1/2)\Delta_t$ | $(n + 1)\Delta_t$ |
| | P_z | $(n + 1)\Delta_t$ | $(n + 1/2)\Delta_t$ |
| B | O_x, O_y | $(n + 1)\Delta_t$ | $(n + 1/2)\Delta_t$ |
| | P_z | $(n + 1/2)\Delta_t$ | $(n + 1)\Delta_t$ |

4 ALTERNATING-DIRECTION IMPLICIT FDTD METHOD

4.1 Algorithm

Recently, the 2D alternating-direction implicit (ADI) FDTD method was proposed [1]. Around the same time other researchers [10] confirmed the appealing property of ADI FDTD, namely unconditional stability. Later the same researchers proposed the 3D ADI FDTD method [11], [12]. This technique is relatively new in electromagnetics but already exists for a long time in other fields, especially computation fluid dynamics.

To derive the time stepping algorithm we first observe the state-space model for a lossless uniform FDTD grid, divided into block matrices for all field components:

$$\begin{bmatrix} \mathbf{D}_{\theta,x} & 0 & 0 \\ 0 & \mathbf{D}_{\theta,y} & 0 \\ 0 & 0 & \mathbf{D}_{\kappa,z} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}}_x \\ \dot{\mathbf{o}}_y \\ \dot{\mathbf{p}}_z \end{bmatrix} = -\frac{c_0}{\Delta} \begin{bmatrix} 0 & 0 & \mathbf{K}_x \\ 0 & 0 & \mathbf{K}_y \\ -\mathbf{K}_x^T & -\mathbf{K}_y^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{p}_z \end{bmatrix} \quad (3.36)$$

The idea of ADI FDTD is to divide the matrix into an x -directed part and a y -directed part:

$$\begin{bmatrix} \mathbf{D}_{\theta,x} & 0 & 0 \\ 0 & \mathbf{D}_{\theta,y} & 0 \\ 0 & 0 & \mathbf{D}_{\kappa,z} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}}_x \\ \dot{\mathbf{o}}_y \\ \dot{\mathbf{p}}_z \end{bmatrix} = -\frac{c_0}{\Delta} \left(\begin{bmatrix} 0 & 0 & \mathbf{K}_x \\ 0 & 0 & 0 \\ -\mathbf{K}_x^T & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{K}_y \\ 0 & -\mathbf{K}_y^T & 0 \end{bmatrix} \right) \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{p}_z \end{bmatrix} \quad (3.37)$$

The skew symmetric matrix on the right is divided into two skew symmetric matrices. By setting

$$\mathbf{x} = \begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{p}_z \end{bmatrix} \quad (3.38a)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \mathbf{K}_x \\ 0 & 0 & 0 \\ -\mathbf{K}_x^T & 0 & 0 \end{bmatrix} \quad (3.38b)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{K}_y \\ 0 & -\mathbf{K}_y^T & 0 \end{bmatrix} \quad (3.38c)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{\theta,x} & 0 & 0 \\ 0 & \mathbf{D}_{\theta,y} & 0 \\ 0 & 0 & \mathbf{D}_{\kappa,z} \end{bmatrix} \quad (3.38d)$$

this can be written shorter as

$$\mathbf{D}\dot{\mathbf{x}} = -\frac{c_0}{\Delta} (\mathbf{A} + \mathbf{B}) \mathbf{x} \quad (3.39)$$

The ADI time stepping algorithm is now [1]:

$$\mathbf{D}(\mathbf{x}|^{n+1/2} - \mathbf{x}|^n) = -\frac{k}{2} (\mathbf{A}\mathbf{x}|^n + \mathbf{B}\mathbf{x}|^{n+1/2}) \quad (3.40a)$$

$$\mathbf{D}(\mathbf{x}|^{n+1} - \mathbf{x}|^{n+1/2}) = -\frac{k}{2} (\mathbf{A}\mathbf{x}|^{n+1} + \mathbf{B}\mathbf{x}|^{n+1/2}) \quad (3.40b)$$

where k is still determined by (3.8). This can be rewritten as:

$$\left(\mathbf{D} + \frac{k}{2}\mathbf{B}\right)\mathbf{x}|^{n+1/2} = \left(\mathbf{D} - \frac{k}{2}\mathbf{A}\right)\mathbf{x}|^n \quad (3.41a)$$

$$\left(\mathbf{D} + \frac{k}{2}\mathbf{A}\right)\mathbf{x}|^{n+1} = \left(\mathbf{D} - \frac{k}{2}\mathbf{B}\right)\mathbf{x}|^{n+1/2} \quad (3.41b)$$

These equations are used in an iterative way and determine the time stepping algorithm for the ADI FDTD method:

1. set $n = 0$ and start with field values equal to zero
2. use (3.41a) to advance time to $t = (n + 1/2)\Delta_t$ for all field values
3. use (3.41b) to advance time to $t = (n + 1)\Delta_t$ for all field values
4. while $n \leq n_{\text{final}}$ increment n and go back to 2

By adding both equations in (3.40):

$$\mathbf{D}(\mathbf{x}|^{n+1} - \mathbf{x}|^n) = -\frac{k}{2}\mathbf{A}(\mathbf{x}|^{n+1} + \mathbf{x}|^n) - k\mathbf{B}\mathbf{x}|^{n+1/2} \quad (3.42)$$

and by considering

$$\dot{\mathbf{x}}|^{n+1/2} = \frac{1}{\Delta_t}(\mathbf{x}|^{n+1} - \mathbf{x}|^n) + \mathcal{O}(\Delta_t^2) \quad (3.43)$$

$$\mathbf{x}|^{n+1/2} = \frac{1}{2}(\mathbf{x}|^{n+1} + \mathbf{x}|^n) + \mathcal{O}(\Delta_t^2) \quad (3.44)$$

it is clear that the two step time stepping in (3.41) is second order accurate.

4.2 Computational considerations

As is illustrated by the term ADI, the method is implicit. This means that the equations in (3.41) are solved for each time step. However the solution of (3.41) is not very complex, this can be shown by writing out (3.41b) in block notation:

$$\begin{bmatrix} \mathbf{D}_{\theta,x} & 0 & \frac{k}{2}\mathbf{K}_x \\ 0 & \mathbf{D}_{\theta,y} & 0 \\ -\frac{k}{2}\mathbf{K}_x^T & 0 & \mathbf{D}_{\kappa,z} \end{bmatrix} \begin{bmatrix} \mathbf{o}_x|^{n+1/2} \\ \mathbf{o}_y|^{n+1/2} \\ \mathbf{p}_z|^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{\theta,x} & 0 & 0 \\ 0 & \mathbf{D}_{\theta,y} & -\frac{k}{2}\mathbf{K}_y \\ 0 & \frac{k}{2}\mathbf{K}_y^T & \mathbf{D}_{\kappa,z} \end{bmatrix} \begin{bmatrix} \mathbf{o}_x|^n \\ \mathbf{o}_y|^n \\ \mathbf{p}_z|^n \end{bmatrix} \quad (3.45)$$

Left multiplying (3.45) by:

$$\begin{bmatrix} \mathbf{I} & 0 & -\frac{k}{2}\mathbf{K}_x\mathbf{D}_{\kappa,z}^{-1} \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \quad (3.46)$$

results in

$$\begin{bmatrix} \mathbf{D}_{\theta,x} + \frac{k^2}{4}\mathbf{K}_x\mathbf{D}_{\kappa,z}^{-1}\mathbf{K}_x^T & 0 & 0 \\ 0 & \mathbf{D}_{\theta,y} & 0 \\ -\frac{k}{2}\mathbf{K}_x^T & 0 & \mathbf{D}_{\kappa,z} \end{bmatrix} \begin{bmatrix} \mathbf{o}_x|^{n+1/2} \\ \mathbf{o}_y|^{n+1/2} \\ \mathbf{p}_z|^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{\theta,x} & \frac{k^2}{4}\mathbf{K}_x\mathbf{D}_{\kappa,z}^{-1}\mathbf{K}_y^T & -\frac{k}{2}\mathbf{K}_x \\ 0 & \mathbf{D}_{\theta,y} & -\frac{k}{2}\mathbf{K}_y \\ 0 & \frac{k}{2}\mathbf{K}_y^T & \mathbf{D}_{\kappa,z} \end{bmatrix} \begin{bmatrix} \mathbf{o}_x|^n \\ \mathbf{o}_y|^n \\ \mathbf{p}_z|^n \end{bmatrix} \quad (3.47)$$

The matrix-vector multiplication on the right is still sparse, although the degree of sparsity has decreased, due to the fact that products of sparse matrices are still sparse and by noting that \mathbf{D} and \mathbf{D}^{-1} are diagonal. The work that is needed to solve the equation is then only determined by

$$\mathbf{D}_{\theta,x} + \frac{k^2}{4}\mathbf{K}_x\mathbf{D}_{\kappa,z}^{-1}\mathbf{K}_x^T \quad (3.48)$$

The non-zero elements, and therefore the computational cost, are equal in the vacuum case:

$$\mathbf{I} + \frac{k^2}{4} \mathbf{K}_x \mathbf{K}_x^T \quad (3.49)$$

and by using for instance (2.61), for the case explained in Section 4.3.1.4 of Chapter 2, it is possible to determine the computational complexity:

$$\mathbf{I} + \frac{k^2}{4} \mathbf{K}_x \mathbf{K}_x^T = \mathbf{I} + \frac{k^2}{4} (\mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y}) (\mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y}^T) \quad (3.50)$$

$$= \mathbf{I} + (\mathbf{I}_{n_x} \mathbf{I}_{n_x}) \otimes \left(\frac{k^2}{4} \mathbf{W}_{n_y} \mathbf{W}_{n_y}^T \right) \quad (3.51)$$

$$= \mathbf{I}_{n_x} \otimes \mathbf{I}_{n_y+1} + \mathbf{I}_{n_x} \otimes \left(\frac{k^2}{4} \mathbf{W}_{n_y} \mathbf{W}_{n_y}^T \right) \quad (3.52)$$

$$= \mathbf{I}_{n_x} \otimes \left(\mathbf{I}_{n_y+1} + \frac{k^2}{4} \mathbf{W}_{n_y} \mathbf{W}_{n_y}^T \right) \quad (3.53)$$

This indicates that solving for $O_x |^{n+1/2}$ in (3.47) only requires the solution of n_x systems of $(n_y + 1)$ linear equations. Furthermore each system that needs to be solved is tridiagonal which results in a very fast solution of (3.41b). Solving (3.41a) can, in a similar way, also be done efficiently. The same reasoning can be applied to show that (3.41a) requires the solution of n_y systems of $(n_x + 1)$ linear equations (see [1]). The ADI FDTD method is not as efficient as the standard FDTD method, but this can be compensated by choosing a larger time step since a restriction on the size of the time step no longer exists.

Remark that we are no longer working with a leapfrog time stepping algorithm since all fields are updated at the same time. The grid however is still interleaved. Furthermore it should be mentioned that it has only been proposed for a uniform orthogonal grid and is thus, at least up to now, not suited in combination with subgridding.

BIBLIOGRAPHY

- [1] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 10, pp. 2003–2007, Oct. 1999.
- [2] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 1995.
- [3] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*, CRC Press, 1993.

- [4] L. Gürel and U. Oğuz, "Signal-processing techniques to reduce the sinusoidal steady-state error in the FDTD method," *IEEE Trans. Antennas Propagat.*, vol. 48, no. 4, pp. 585–593, Apr. 2000.
- [5] A. P. Zhao and A. V. Räisänen, "Application of a simple and efficient source excitation technique to the FDTD analysis of waveguide and microstrip circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, no. 9, pp. 1535–1539, Sep. 1996.
- [6] M. Okoniewski, E. Okoniewska, and M. A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 3, pp. 422–429, Mar. 1997.
- [7] K. M. Krishnaiah and C. J. Railton, "A stable subgridding algorithm and its application to eigenvalue problems," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 5, pp. 620–628, May 1999.
- [8] D. T. Prescott and N. V. Shuley, "A method for incorporating different sized cells into the finite-difference time-domain analysis technique," *IEEE Microwave and Guided Wave Letters*, vol. 2, no. 11, pp. 434–436, Nov. 1992.
- [9] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," *IEEE Microwave and Guided Wave Letters*, vol. 10, no. 8, pp. 301–303, Aug. 2000.
- [10] F. Zheng, Z. Chen, and J. Zhang, "A finite-difference time-domain method without the courant stability conditions," *IEEE Microwave and Guided Wave Letters*, vol. 9, no. 11, pp. 441–443, Nov. 1999.
- [11] T. Namiki, "3-D ADI-FDTD method—unconditionally stable time-domain algorithm for solving full vector maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 10, pp. 1743–1748, Oct. 2000.
- [12] F. Zheng, Z. Chen, and J. Zhang, "Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1550–1558, Sep. 2000.

CHAPTER 4

ROM: Reduced order modeling

1 INTRODUCTION

Reduced order modeling algorithms have, over the last years, grown significantly in importance in the field of circuit modeling. This is primarily due to the growing complexity of integrated electronic circuits, and due to the more complex, circuit-based, modeling techniques, e.g. the PEEC-technique (Partial Element Equivalent Circuit), required to incorporate parasitic effects of high-speed electronics. In [1], the problem, that boosted the interest in reduced order modeling (ROM) techniques, was formulated as follows:

Today's integrated circuits are extremely complex, with up to hundreds of thousands or even millions of devices, and prototyping of such circuits is no longer possible. Instead, computational methods are used to simulate and analyze the behavior of the electronic circuit at the design stage. This allows to correct the design before the circuit is actually fabricated in silicon. Today's trends in VLSI technology have brought to light a number of challenging simulation problems for which traditional circuit-simulation techniques are no longer adequate. In particular, due to ever-increasing clock rates and ever-decreasing feature sizes, the accurate simulation of the wires that connect the functional blocks of a circuit, the so-called *interconnect*, has become a crucial issue. At the chip level, an interconnect can be modeled by large, lumped, linear networks that are generated by automatic parasitics-extraction programs. Typically, these networks are

too large to be included directly in the simulation of the complete circuit. Instead, the large networks are replaced by much smaller reduced-order models that are then integrated into the general design and simulation methodology.

The higher frequencies and smaller dimensions cause the modeling of interconnects to be more complex due to parasitic coupling effects [2]. High speed effects as delay, rise time degradation, attenuation, crosstalk, skin effect, reflection, ... have to be considered and have to be taken into account. Since the circuits consist of linear and nonlinear parts where a linear part is, e.g. the extracted model of the interconnect based on the partial element equivalent circuit (PEEC) technique and where a nonlinear part contains the transistors and diodes inside the circuit, these parts have to be isolated from each other. Employing a reduced order modeling technique, the linear parts are then approximated by a smaller system having similar behaviour in a specific sense. This reduced system is then recombined with the nonlinear parts to allow the analysis of the original circuit.

The use of these ROM techniques is not only restricted to circuit modeling, but can also be used in combination with the FDTD technique. This was already mentioned in Section 3.3 of Chapter 3, where an implicit time stepping algorithm was introduced. This time stepping algorithm is not useful unless a reduced order modeling (ROM) algorithm is used to limit the number of variables needed to describe the behaviour of the subdomain.

In this chapter some aspects concerning reduced order modeling will be treated. As mentioned above, the interest in ROM techniques has grown strongly during the last years. It is not our intention to give an overview of all possible techniques but we will restrict ourselves to the most important techniques and steps that have lead to the development of a specific technique [3] which will, here, be referred to as the *Laguerre-SVD* ROM technique. The numerical examples, to be discussed in Chapter 6, all use this ROM technique.

The chapter is constructed as follows: in Section 2 some general aspects concerning ROM are investigated. In Section 3 a number of ROM algorithms are discussed in more detail. In particular the Padé via Lanczos (PVL), the Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA) and the Laguerre-SVD technique, will be discussed. Finally in Section 4 the relation between FDTD and ROM will be discussed. First of all by looking back at what can be found in the literature, followed by a look at the subdomain based algorithms introduced in this work. The section will conclude with the final algorithms.

2 REDUCED ORDER MODELING: GENERAL ASPECTS

Let us start by considering the state-space model as it was introduced in Chapter 2:

$$\begin{cases} \mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{L}^T \mathbf{x} \end{cases} \quad (4.1)$$

This state-space model is a set of first-order differential equations. It describes the behaviour of the output variables \mathbf{y} as a function of the input variables \mathbf{u} by means of a number of internal variables \mathbf{x} . The dimensions of the different vectors are: $\dim[\mathbf{x}] = N$, $\dim[\mathbf{u}] = p$ and $\dim[\mathbf{y}] = m$. The matrices \mathbf{G} and \mathbf{C} then have dimension $N \times N$, the dimension of matrix \mathbf{B} is $N \times p$ and the dimension of \mathbf{L} is $N \times m$. When $p, m \ll N$, ROM can be used to generate a smaller model having similar behaviour.

Using a ROM technique, the original system (4.1) can be approximated by a state-space model of reduced size:

$$\begin{cases} \tilde{\mathbf{C}}\dot{\mathbf{z}} = -\tilde{\mathbf{G}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} \\ \mathbf{y} = \tilde{\mathbf{L}}^T \mathbf{z} \end{cases} \quad (4.2)$$

The vectors \mathbf{u} and \mathbf{y} can still be distinguished, but the vector \mathbf{x} containing the internal variables has been replaced by a new smaller vector \mathbf{z} ($\dim[\mathbf{z}] < \dim[\mathbf{x}]$). The matrices in (4.2) have correspondingly smaller dimensions: $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{C}}$ have dimension $\dim[\mathbf{z}] \times \dim[\mathbf{z}]$, $\tilde{\mathbf{B}}$ is of dimension $\dim[\mathbf{z}] \times p$ and $\tilde{\mathbf{L}}$ is of dimension $\dim[\mathbf{z}] \times m$. The reduced state-space model still describes the behaviour of the output variables \mathbf{y} as a function of the input variables \mathbf{u} , however by means of a smaller number of new internal variables \mathbf{z} , hence a model of reduced order. The ROM techniques make sure that this new, smaller, state-space model (4.2) is a good approximation of the original one (4.1) in some sense.

Since we are trying to incorporate the reduced model in FDTD simulations, a technique which involves only operations on real numbers, we would like the matrices $\tilde{\mathbf{C}}$, $\tilde{\mathbf{G}}$, $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{L}}$ describing the reduced model to be real as well.

With zero initial conditions and unit impulse excitations, the transfer matrix belonging to (4.1) describing the input-output relation in the Laplace domain is given by:

$$\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{L}^T (\mathbf{G} + s\mathbf{C})^{-1} \mathbf{B} \quad (4.3)$$

and for the reduced system (4.2) this transfer matrix is:

$$\tilde{\mathbf{H}}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \tilde{\mathbf{L}}^T (\tilde{\mathbf{G}} + s\tilde{\mathbf{C}})^{-1} \tilde{\mathbf{B}} \quad (4.4)$$

Each of the ROM techniques starts by introducing a change of variables of the following form:

$$s = \frac{a\sigma + b}{c\sigma + d} \quad \sigma = -\frac{b - sd}{a - sc} \quad (4.5)$$

where the following condition must hold:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} \neq 0 \quad (4.6)$$

and $a, b, c, d \in \mathbb{R}$. This change of variables is called a bilinear or Möbius transformation, it has the property to transform generalized circles into generalized circles [4]. A generalized circle can be any circle including straight lines, which are circles with infinite radius. This change of variables has a similar effect on both transfer matrices:

$$\mathbf{H}(\sigma) = (c\sigma + d)\mathbf{L}^T (\mathbf{I} - \sigma\mathbf{A})^{-1} \mathbf{R} \quad (4.7a)$$

$$\tilde{\mathbf{H}}(\sigma) = (c\sigma + d)\tilde{\mathbf{L}}^T (\mathbf{I} - \sigma\tilde{\mathbf{A}})^{-1} \tilde{\mathbf{R}} \quad (4.7b)$$

where the new matrices \mathbf{A} and \mathbf{R} are related to \mathbf{C} , \mathbf{G} and \mathbf{B} as:

$$\mathbf{A} = -(d\mathbf{G} + b\mathbf{C})^{-1}(c\mathbf{G} + a\mathbf{C}) \quad (4.8)$$

$$\mathbf{R} = (d\mathbf{G} + b\mathbf{C})^{-1}\mathbf{B} \quad (4.9)$$

and similar for $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{R}}$. The matrix \mathbf{I} is the identity matrix and has the appropriate dimensions. The function $(c\sigma + d)$ does not play an important part in the rest of the explanation, making sure that, in some kind of form

$$\tilde{\mathbf{L}}^T (\mathbf{I} - \sigma\tilde{\mathbf{A}})^{-1} \tilde{\mathbf{R}} \quad (4.10)$$

is a good approximation for

$$\mathbf{L}^T (\mathbf{I} - \sigma\mathbf{A})^{-1} \mathbf{R} \quad (4.11)$$

ensuring that the transfermatrix (4.4) is a good approximation for (4.3).

Based on the matrix identity (Neumann Series) [5]:

$$(\mathbf{I} - \mathbf{E})^{-1} = \sum_{i=0}^{\infty} \mathbf{E}^i \quad \text{whenever} \quad \rho(\mathbf{E}) < 1 \quad (4.12)$$

where $\rho(\mathbf{E}) = \max \{|\lambda| : \lambda \text{ is an eigenvalue of } \mathbf{E}\}$ is the spectral radius of \mathbf{E} , it is possible to explain in what sense the reduced order models are an approximation of the original ones. By applying (4.12) to (4.7a), we can write the transfer matrix as a Taylor expansion around $\sigma = 0$:

$$\mathbf{H}(\sigma) = (c\sigma + d)\mathbf{L}^T \left(\sum_{i=0}^{\infty} \mathbf{A}^i \sigma^i \right) \mathbf{R} \quad (4.13)$$

$$= (c\sigma + d) \sum_{i=0}^{\infty} \mathbf{L}^T \mathbf{A}^i \mathbf{R} \sigma^i \quad (4.14)$$

$$= (c\sigma + d) \sum_{i=0}^{\infty} \mathbf{M}_i \sigma^i \quad (4.15)$$

valid for small σ , $|\sigma| < 1/\rho(\mathbf{A})$. The matrices \mathbf{M}_i are called the *moments* or the *Markov parameters*. For the transfer matrix of the reduced system this Taylor expansion becomes:

$$\tilde{\mathbf{H}}(\sigma) = (c\sigma + d)\tilde{\mathbf{L}}^T \left(\sum_{i=0}^{\infty} \tilde{\mathbf{A}}^i \sigma^i \right) \tilde{\mathbf{R}} \quad (4.16)$$

$$= (c\sigma + d) \sum_{i=0}^{\infty} \tilde{\mathbf{L}}^T \tilde{\mathbf{A}}^i \tilde{\mathbf{R}} \sigma^i \quad (4.17)$$

$$= (c\sigma + d) \sum_{i=0}^{\infty} \tilde{\mathbf{M}}_i \sigma^i \quad (4.18)$$

The idea is to match the moments of the reduced system with the moments of the original system:

$$\mathbf{M}_i = \mathbf{L}^T \mathbf{A}^i \mathbf{R} = \tilde{\mathbf{L}}^T \tilde{\mathbf{A}}^i \tilde{\mathbf{R}} = \tilde{\mathbf{M}}_i \quad \text{for } 0 \leq i < q \quad (4.19)$$

making sure that in this way

$$\mathbf{H}(\sigma) = \tilde{\mathbf{H}}(\sigma) + (c\sigma + d)\mathcal{O}(\sigma^q) \quad (4.20)$$

In this chapter a number of ROM algorithms will be studied. In the Padé via Lanczos (PVL) technique [6–8] the choice for σ is:

$$\sigma = s - s_0 \quad (4.21)$$

or

$$a = 1 \quad (4.22a)$$

$$b = s_0 \quad (4.22b)$$

$$c = 0 \quad (4.22c)$$

$$d = 1 \quad (4.22d)$$

and the matrices \mathbf{A} and \mathbf{R} become:

$$\mathbf{A} = -(\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C} \quad \mathbf{R} = (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{B} \quad (4.23)$$

In this way a Taylor expansion about $s = s_0$ is obtained.

In the Passive Reduced order Interconnect Macromodeling Algorithm (PRIMA) technique [9], this expansion point is set to zero, $s_0 = 0$:

$$\sigma = s \quad (4.24)$$

or

$$a = 1 \quad (4.25a)$$

$$b = 0 \quad (4.25b)$$

$$c = 0 \quad (4.25c)$$

$$d = 1 \quad (4.25d)$$

and thus

$$\mathbf{A} = -\mathbf{G}^{-1}\mathbf{C} \quad \mathbf{R} = \mathbf{G}^{-1}\mathbf{B} \quad (4.26)$$

When the state-space model has been obtained using the FDTD technique [10–12] it is known that \mathbf{C} is non-singular and diagonal. This property can be exploited by choosing:

$$\sigma = \frac{1}{s} \quad (4.27)$$

or

$$a = 0 \quad (4.28a)$$

$$b = 1 \quad (4.28b)$$

$$c = 1 \quad (4.28c)$$

$$d = 0 \quad (4.28d)$$

and where as a consequence \mathbf{A} and \mathbf{R} are easy to calculate:

$$\mathbf{A} = -\mathbf{C}^{-1}\mathbf{G} \quad \mathbf{R} = \mathbf{C}^{-1}\mathbf{B} \quad (4.29)$$

Since the approximate model is only valid for σ small, this corresponds to an expansion point $s = \infty$. The implications of this choice will be examined in Section 4.

A final ROM algorithm that will be studied is called the Laguerre-SVD algorithm [3, 13]. There the choice for σ is:

$$\sigma = \frac{s - \alpha}{s + \alpha} \quad (4.30)$$

or

$$a = \alpha \quad (4.31a)$$

$$b = \alpha \quad (4.31b)$$

$$c = -1 \quad (4.31c)$$

$$d = 1 \quad (4.31d)$$

which results in:

$$\mathbf{A} = -(\alpha \mathbf{C} + \mathbf{G})^{-1}(\alpha \mathbf{C} - \mathbf{G}) \quad \mathbf{R} = (\alpha \mathbf{C} + \mathbf{G})^{-1} \mathbf{B} \quad (4.32)$$

All the ROM techniques discussed here involve Krylov subspaces, where the q -th block Krylov subspace induced by \mathbf{A} and \mathbf{R} is defined as:

$$\mathcal{K}(\mathbf{A}, \mathbf{R}, q) = \text{span}\{\mathbf{R}, \mathbf{A}\mathbf{R}, \mathbf{A}^2\mathbf{R}, \dots, \mathbf{A}^{q-1}\mathbf{R}\} \quad (4.33)$$

and the block Krylov subspace induced by \mathbf{A}^T and \mathbf{L} is defined as:

$$\mathcal{K}(\mathbf{A}^T, \mathbf{L}, q) = \text{span}\left\{\mathbf{L}, \mathbf{A}^T\mathbf{L}, (\mathbf{A}^T)^2\mathbf{L}, \dots, (\mathbf{A}^T)^{q-1}\mathbf{L}\right\} \quad (4.34)$$

The importance of these Krylov subspaces lies in the fact that they contain all the information related to the first q moments \mathbf{M}_i . Most ROM algorithms generate a basis for one or both subspaces and employ this basis, either explicitly, through the use of projection-based techniques, or implicitly, through the use of Krylov subspace techniques as for example the Lanczos or the Arnoldi algorithm, to generate a reduced model. This will become more clear in the following section.

Other ROM techniques, not based on Krylov subspace methods, were introduced in the past. The most famous of these is the balanced realizations technique [14]. In the balanced realizations technique, the gain between input and states on the one hand, and states and output on the other hand, is first balanced. In a next step the states with lowest gain are removed, thus obtaining a ROM. However, as illustrated in [15], even in the, for the balanced realization favorable, symmetric case, the accuracy of the balanced realization technique is lower and the computation time is higher as compared to the Laguerre-SVD technique [3]. Therefore this technique will not be considered here.

3 REDUCED ORDER MODELING ALGORITHMS

3.1 PVL: Padé via Lanczos

Prior to the introduction of the Padé Via Lanczos (PVL) method, the Asymptotic Waveform Evaluation (AWE) was introduced [16]. As is also the case

in PVL, a Padé approximation of the transfer function is generated in the AWE technique. Since only single-input single-output systems ($m = p = 1$) are considered, the transfer matrix is actually a function. In a first step the moments are calculated explicitly and in a second step a Padé approximant is derived from these moments. The main problem concerning this straightforward approach was the explicit use of the moments and the ill-conditioned computations resulting from it [6]. To understand this, it has to be noted that expressions such as $\mathbf{A}^i \mathbf{r}$ converge to an eigenvector corresponding to the largest eigenvalue belonging to \mathbf{A} as i increases. Therefore the moments predominantly contain information belonging to a single eigenvalue. This resulted in ROM's for which the transfer function did not converge to the original transfer function as the complexity of the reduced model increased. From a certain point on, increasing the order of approximation q , resulting in more complex reduced systems, did not improve the transfer function of the reduced system.

As a reaction in 1995, the Padé via Lanczos ROM technique was introduced [6]. It promised to reduce an original, single-input single-output (SISO) system ($m = p = 1$) as accurately as desired: the higher the order of the reduced system is, or equivalently the higher the number of the new internal variables is, the better the approximation of the ROM becomes. This statement proved right thanks to the use of the well conditioned Lanczos process, which is a Krylov subspace technique. Hence the name of the algorithm, Padé via Lanczos: generate a Padé approximant or a Padé-type approximant by means of the Lanczos process.

For a SISO system, where \mathbf{R} and \mathbf{B} become a vector and are denoted by small letters, the PVL algorithm introduces first of all the new variable $\sigma = s - s_0$, and for the system characterized by:

$$\mathbf{A} = -(\mathbf{G} + s_0 \mathbf{C})^{-1} \mathbf{C} \quad \mathbf{r} = (\mathbf{G} + s_0 \mathbf{C})^{-1} \mathbf{b} \quad (4.35)$$

the Lanczos process is used to tridiagonalize \mathbf{A} :

$$\mathbf{W}^T \mathbf{A} \mathbf{V} = \mathbf{D} \mathbf{T} \quad (4.36)$$

where \mathbf{T} is a real tridiagonal matrix and

$$\mathbf{D} = \mathbf{W}^T \mathbf{V} \quad (4.37)$$

is a real, diagonal matrix indicating that the vectors of \mathbf{W} and \mathbf{V} are biorthogonal. Both are of dimension $q \times q$. The columns of \mathbf{V} , form a basis for the right Krylov subspace (4.33):

$$\text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_q\} = \mathcal{K}(\mathbf{A}, \mathbf{R}, q) \quad (4.38)$$

and the columns of \mathbf{W} a basis for the left Krylov subspace (4.34):

$$\text{span}\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_q\} = \mathcal{K}(\mathbf{A}^T, \mathbf{L}, q) \quad (4.39)$$

The matrices \mathbf{V} , \mathbf{W} are of dimension $N \times q$. The original transfer function:

$$\mathbf{H}(\sigma) = \mathbf{I}^T (\mathbf{I} - \sigma \mathbf{A})^{-1} \mathbf{r} \quad (4.40)$$

can be approximated by the transfer function of a reduced system:

$$\tilde{\mathbf{H}}(\sigma) = \mathbf{I}^T \mathbf{V} (\mathbf{W}^T \mathbf{V} - \sigma \mathbf{W}^T \mathbf{A} \mathbf{V})^{-1} \mathbf{W} \mathbf{r} \quad (4.41)$$

$$= \mathbf{I}^T \mathbf{V} (\mathbf{I} - \sigma \mathbf{T})^{-1} \mathbf{D}^{-1} \mathbf{W} \mathbf{r} \quad (4.42)$$

where the dimension of the matrices is now q , where it used to be N . It was shown in [6], that $2q$ moments are matched in this way.

Later this algorithm was extended to systems with multiple inputs and multiple outputs ($p \neq 1$ and $m \neq 1$) [7] and was called the Matrix Padé Via Lanczos (MPVL) algorithm. The number of moments matched by MPVL is also maximum [8]. For $m = p$, a system of size $qp = qm$ matches $2q$ moments.

The complexity of the Lanczos algorithm increases strongly when effects as *deflation* and *breakdown* need to be covered. The first effect, deflation, occurs when a vector of $\mathbf{A}^i \mathbf{R}$ is linearly dependent on the vectors already present in the right Krylov matrix. The deflation condition is met by removing the corresponding vectors from $\mathbf{A}^i \mathbf{R}$ before continuing the calculation of $\mathbf{A}(\mathbf{A}^i \mathbf{R})$, and the same for $(\mathbf{A}^T)^i \mathbf{L}$. A second problem occurring with PVL and MPVL is breakdown. The Lanczos process breaks down when division by zero is performed before the full basis of the Krylov subspace has been constructed. This problem is resolved by introducing a look-ahead step, which relaxes the vector-wise biorthogonality (4.37) to a cluster-wise biorthogonality, when division by zero, originating from $\mathbf{v}_i^T \mathbf{w}_i \simeq 0$, is detected.

The expansion point, s_0 , needs to be chosen as well. This expansion point needs to be real, since we would like the ROM to be described by real matrices. In [6], for problems where the frequency range of interest is $0 \leq f < f_{\max}$, the choice

$$s_0 = 2\pi f_{\max} \quad (4.43)$$

for the expansion point was proposed.

One of the strengths of the (M)PVL algorithm is the low memory requirements related to the process. For the single-input single-output case, only \mathbf{D} , \mathbf{T} , \mathbf{A} and the vectors \mathbf{v}_{q+1} , \mathbf{v}_q , \mathbf{v}_{q-1} , \mathbf{w}_{q+1} , \mathbf{w}_q and \mathbf{w}_{q-1} need to be stored in memory as the process continues. In other words, it is not necessary to

store \mathbf{V}_q and \mathbf{W}_q entirely. An other strength is the high order, measured in number of moments, of the Padé approximations generated by PVL or MPVL.

A serious weakness, however, related to the PVL process is the lack of stability associated with these reduced models. In [8], an example is presented where a reduced model is generated using a symmetric version of the matrix Padé via Lanczos algorithm (SyMPVL), that produced a very accurate model having however a few poles in the right half of the complex plane.

Since we would like to use the reduced model in an iterative time stepping algorithm, see (3.28), we cannot afford this. A possible solution to circumvent this is to remove the unstable poles from the model and to hope that these poles do not have a major role. Fortunately, there is another solution, first observed in [9] and used together with the Arnoldi process and later applied to the PVL process in [8]. It is applicable to systems where $\mathbf{B} = \mathbf{L}$.

The main idea is to generate a reduced model, by projecting the original linear dynamical system (4.1) onto the Krylov subspace $\mathcal{K}(\mathbf{A}, \mathbf{R}, q)$. This is done by employing \mathbf{V} , a basis for $\mathcal{K}(\mathbf{A}, \mathbf{R}, q)$. The matrices of the ROM (4.2) are then given by:

$$\tilde{\mathbf{C}} = \mathbf{V}^T \mathbf{C} \mathbf{V} \quad (4.44a)$$

$$\tilde{\mathbf{G}} = \mathbf{V}^T \mathbf{G} \mathbf{V} \quad (4.44b)$$

$$\tilde{\mathbf{B}} = \mathbf{V}^T \mathbf{B} \quad (4.44c)$$

One basic difference, between ROM's obtained in this way and obtained using the original PVL method, is that, here, only one Krylov subspace is involved.

However, the accuracy, in terms of moments, of the ROM's generated in this way is not as high as it is for the ROM generated by means of the Lanczos process (4.42). There are only q moments matched, when a basis for $\text{span}\{\mathbf{V}\} = \mathcal{K}(\mathbf{A}, \mathbf{R}, q)$ is used. Which is only half as much as before. The moments related to the new transfer function (4.4) can be written as

$$\tilde{\mathbf{M}}_i = \tilde{\mathbf{L}}^T \tilde{\mathbf{A}}^i \tilde{\mathbf{R}} = \tilde{\mathbf{L}}^T \left[-(\tilde{\mathbf{G}} + s_0 \tilde{\mathbf{C}})^{-1} \tilde{\mathbf{C}} \right]^i \tilde{\mathbf{R}} \quad (4.45)$$

$$= \mathbf{L}^T \left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0 \tilde{\mathbf{C}})^{-1} \mathbf{V}^T \mathbf{C} \right]^i \mathbf{V} \tilde{\mathbf{R}} \quad (4.46)$$

and the moments related to the original system are:

$$\mathbf{M}_i = \mathbf{L}^T \mathbf{A}^i \mathbf{R} = \mathbf{L}^T \left[-(\mathbf{G} + s_0 \mathbf{C})^{-1} \mathbf{C} \right]^i \mathbf{R} \quad (4.47)$$

these are equal for $i = 0, 1, \dots, (q - 1)$. This can be shown in two steps: in a first step the equality

$$\mathbf{V}\tilde{\mathbf{R}} = \mathbf{R} \quad (4.48)$$

is shown and in a second step,

$$\left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \right]^i \mathbf{R} = \left[-(\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C} \right]^i \mathbf{R} \quad (4.49)$$

$$= \mathbf{A}^i \mathbf{R} \quad (4.50)$$

for $i = 0, 1, \dots, q - 1$, is shown.

First it is observed, since \mathbf{V} is a basis for the right Krylov subspace (4.33), that a matrix \mathbf{E}_i , $i = 0, 1, \dots, q - 1$, exists such that

$$\mathbf{A}^i \mathbf{R} = \mathbf{V}\mathbf{E}_i \quad (4.51)$$

More specifically, for $i = 0$ this results in:

$$\mathbf{R} = (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{B} = \mathbf{V}\mathbf{E}_0 \quad (4.52)$$

and left multiplying by $\mathbf{V}^T(\mathbf{G} + s_0\mathbf{C})$ gives:

$$\mathbf{V}^T\mathbf{B} = \tilde{\mathbf{B}} = \mathbf{V}^T(\mathbf{G} + s_0\mathbf{C})\mathbf{V}\mathbf{E}_0 \quad (4.53)$$

$$= (\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})\mathbf{E}_0 \quad (4.54)$$

Left multiplying this result with $\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}$

$$\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\tilde{\mathbf{B}} = \mathbf{V}\tilde{\mathbf{R}} = \mathbf{V}\mathbf{E}_0 = \mathbf{R} \quad (4.55)$$

results in (4.48), proving the first step.

Equation (4.50) can be shown iteratively. When, for $0 \leq i < (q - 1)$,

$$\left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \right]^i \mathbf{R} = \mathbf{A}^i \mathbf{R} \quad (4.56)$$

holds, we will show it also holds for $i + 1$. Left multiplying with \mathbf{A} and making use of (4.51) we get:

$$-(\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C} \left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \right]^i \mathbf{R} = \mathbf{A}^{i+1}\mathbf{R} = \mathbf{V}\mathbf{E}_{i+1} \quad (4.57)$$

and multiplying by $\mathbf{V}^T(\mathbf{G} + s_0\mathbf{C})$ results in:

$$-\mathbf{V}^T\mathbf{C} \left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \right]^i \mathbf{R} = \mathbf{V}^T(\mathbf{G} + s_0\mathbf{C})\mathbf{V}\mathbf{E}_{i+1} \quad (4.58)$$

$$= (\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})\mathbf{E}_{i+1} \quad (4.59)$$

Left multiplying a last time with $\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}$

$$-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \right]^i \mathbf{R} \quad (4.60)$$

$$= \left[-\mathbf{V}(\tilde{\mathbf{G}} + s_0\tilde{\mathbf{C}})^{-1}\mathbf{V}^T\mathbf{C} \right]^{i+1} \mathbf{R} \quad (4.61)$$

$$= \mathbf{V}\mathbf{E}_{i+1} \quad (4.62)$$

$$= \mathbf{A}^{i+1}\mathbf{R} \quad (4.63)$$

shows the second step (4.50).

Despite the loss of accuracy, when the same size of the ROM is considered, there is an important property related to the reduced system with matrices (4.44): the stability of these ROM's, when starting from a system complying with a number of conditions, can be preserved based on the *passivity* of the original and reduced system. Passivity is a property only related to circuits and therefore not obviously translatable to vector fields. However, as will be shown in the following section, the original matrices, as obtained from the FDTD method, are strongly related to the matrices resulting from RLC circuits. For this reason we focus on this kind of ROM technique.

3.2 PRIMA: Passive Reduced-Order Interconnect Macro-modeling Algorithm

In [9] a provably passive ROM technique was proposed for the first time: Passive Reduced-Order Interconnect Macromodeling Algorithm or, in short, PRIMA. It is based on the observation that the state-space model of an RLC circuit, where only resistors, inductors and capacitors are present, can be written, when the modified nodal analysis (MNA) is used [17], as follows:

$$\begin{cases} \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \mathbf{i} \end{bmatrix} = - \begin{bmatrix} \mathbf{N} & \mathbf{E} \\ -\mathbf{E}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{B}^T \begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix} \end{cases} \quad (4.64)$$

where \mathbf{N} , \mathbf{Q} and \mathbf{H} contain the stamps for the resistors, capacitors and inductors respectively. The matrix \mathbf{E} consists of ones, minus ones and zeros, representing the Kirchoff current law equations. Furthermore the matrices \mathbf{N} , \mathbf{Q} and \mathbf{H} are symmetric positive semidefinite matrices. For a system that can be expressed as (4.64), it can be proven that the reduced system, generated using the PRIMA algorithm is passive as well.

In the PRIMA algorithm the block Arnoldi process, a Krylov subspace technique, is used to generate an orthonormal basis (\mathbf{X}) for $\mathcal{K}(\mathbf{A}, \mathbf{R}, q)$, where, since there is no change of variables, hence $\sigma = s$, we have:

$$\mathbf{A} = -\mathbf{G}^{-1}\mathbf{C} \quad \text{and} \quad \mathbf{R} = \mathbf{G}^{-1}\mathbf{B} \quad (4.65)$$

This orthonormal basis:

$$\text{span}(\mathbf{X}) = \mathcal{K}(\mathbf{A}, \mathbf{R}, q) \quad (4.66)$$

is then used to construct the matrices of the reduced system (4.2):

$$\tilde{\mathbf{C}} = \mathbf{X}^T \mathbf{C} \mathbf{X} \quad (4.67a)$$

$$\tilde{\mathbf{G}} = \mathbf{X}^T \mathbf{G} \mathbf{X} \quad (4.67b)$$

$$\tilde{\mathbf{B}} = \mathbf{X}^T \mathbf{B} \quad (4.67c)$$

This is very similar to (4.44), and indeed in [9] it is shown that the first q moments are equal.

The passivity property is preserved, thanks to the structure of the original matrices, \mathbf{C} and \mathbf{G} , used in (4.67). For passivity the following property is required:

$$\mathbf{C}^T + \mathbf{C} \geq 0 \quad (4.68a)$$

$$\mathbf{G}^T + \mathbf{G} \geq 0 \quad (4.68b)$$

where \geq refers to the positive semidefiniteness of a matrix, and this property is preserved in the reduced matrices:

$$\tilde{\mathbf{C}}^T + \tilde{\mathbf{C}} = \mathbf{V}^T (\mathbf{C}^T + \mathbf{C}) \mathbf{V} \geq 0 \quad (4.69a)$$

$$\tilde{\mathbf{G}}^T + \tilde{\mathbf{G}} = \mathbf{V}^T (\mathbf{G}^T + \mathbf{G}) \mathbf{V} \geq 0 \quad (4.69b)$$

The difference between the projection based PVL algorithm (4.44) and the PRIMA algorithm can be observed in a number of points.

- Since the expansion point is set to zero ($s_0 = 0$) in PRIMA, the PVL algorithm is more general ($s_0 \neq 0$). This implies that the PRIMA algorithm requires the matrix \mathbf{G} to be nonsingular. The PVL algorithm, on the other hand, only requires the matrix $(\mathbf{G} + s_0 \mathbf{C})$ to be nonsingular, which can be assured by appropriately choosing s_0 .
- The algorithm employed to calculate the basis, used for the projection, is the Lanczos algorithm in PVL (i.e. for matrix \mathbf{V}) and the Arnoldi algorithm in PRIMA (i.e. for matrix \mathbf{X}). The Arnoldi algorithm generates an orthonormal basis ($\mathbf{X}^T \mathbf{X} = \mathbf{I}$), whereas the Lanczos algorithm generates a normalized basis, but the columns of \mathbf{V} are not necessarily orthogonal. It is numerically more interesting to choose the orthogonal basis of PRIMA.

- Computationally, the Lanczos process is more interesting since, when $m = p = 1$, \mathbf{v}_{i+1} can be calculated from \mathbf{v}_i and \mathbf{v}_{i-1} . For the Arnoldi process \mathbf{x}_{i+1} has to be orthogonalized to $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i$. When memory requirements are considered, both algorithms are comparable since both algorithms have to store the entire \mathbf{V} or \mathbf{X} matrix because they are needed for the construction of $\tilde{\mathbf{C}}$, $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{B}}$.

3.3 Laguerre-SVD reduced order modeling

Finally we consider the Laguerre-SVD ROM algorithm [3, 13]. In contrast to the previous methods the change of variables here is not linear:

$$\sigma = \frac{s - \alpha}{s + \alpha} \quad (4.70)$$

The change of variables maps the s -domain Laguerre expansion

$$\mathbf{H}(s) = \mathbf{L}^T (\mathbf{G} + s\mathbf{C})^{-1} \mathbf{B} = \frac{\sqrt{2\alpha}}{s + \alpha} \sum_{i=0}^{\infty} \mathbf{M}_i \left(\frac{s - \alpha}{s + \alpha} \right)^i \quad (4.71)$$

to the σ -domain power expansion, familiar from previous ROM techniques:

$$\mathbf{H}(\sigma) = (1 - \sigma) \mathbf{L}^T ((\alpha \mathbf{C} + \mathbf{G}) + \sigma(\alpha \mathbf{C} - \mathbf{G}))^{-1} \mathbf{B} \quad (4.72)$$

$$= \frac{1 - \sigma}{\sqrt{2\alpha}} \sum_{i=0}^{\infty} \mathbf{M}_i \sigma^i \quad (4.73)$$

The functions related to the moments \mathbf{M}_i :

$$\frac{\sqrt{2\alpha}}{s + \alpha} \left(\frac{s - \alpha}{s + \alpha} \right)^i \quad (4.74)$$

correspond to the Laplace transformed scaled Laguerre functions:

$$\phi_i^\alpha = \sqrt{2\alpha} e^{-\alpha t} l_i(2\alpha t) \quad (4.75)$$

where α is a scaling parameter that can be chosen and $l_i(t)$ is the Laguerre polynomial

$$l_i(t) = \frac{e^t}{i!} \frac{d^i}{dt^i} (e^{-t} t^i) \quad (4.76)$$

These functions are, in contrast to the power series s^i , a set of orthonormal functions, which is a more sound approach to start from.

By changing the variable to σ , it can be seen that the moments related to:

$$\mathbf{L}^T ((\alpha \mathbf{C} + \mathbf{G}) + \sigma(\alpha \mathbf{C} - \mathbf{G}))^{-1} \mathbf{B} = \frac{1}{\sqrt{2\alpha}} \sum_{i=0}^{\infty} \mathbf{M}_i \sigma^i \quad (4.77)$$

are proportional to the moments belonging to the approximation in Laguerre polynomials of the original transfer matrix (4.71). The change of variables (4.70), has transformed the problem of generating an approximation in Laguerre polynomials, to a problem of generating a Padé approximant or a Padé-type approximant. This last problem, we know how to solve, thanks to the ROM techniques discussed in previous sections. In the Laguerre-SVD technique, the following steps are based on the PRIMA approach. Setting

$$\mathbf{A} = -(\alpha\mathbf{C} + \mathbf{G})^{-1}(\alpha\mathbf{C} - \mathbf{G}) \quad \mathbf{R} = (\alpha\mathbf{C} + \mathbf{G})^{-1}\mathbf{B} \quad (4.78)$$

the reduced matrices can be obtained in a similar manner. However a modification to the PRIMA algorithm has been made. To generate a basis for the Krylov subspace $\mathcal{K}(\mathbf{A}, \mathbf{R}, q)$, the singular value decomposition (SVD) is proposed instead of the Arnoldi process. The SVD also generates an orthonormal basis for Krylov subspace and has proven to be very robust. Although the orthonormal basis obtained with the Arnoldi process, say \mathbf{X} , is probably not equal to the orthonormal basis resulting from the SVD, it can be shown that this has no importance. Even more so, it is easy to show that any basis of the Krylov subspace will do: any new basis \mathbf{U} will always be related to the basis generated by the Arnoldi algorithm \mathbf{X} by some non-singular matrix, say \mathbf{Y} :

$$\mathbf{U} = \mathbf{X}\mathbf{Y} \quad (4.79)$$

where \mathbf{U} and \mathbf{X} have dimension $N \times pq$ and \mathbf{Y} is of dimension $pq \times pq$. The matrices of the reduced system generated using \mathbf{U} instead of \mathbf{X} are then:

$$\begin{aligned} \tilde{\mathbf{C}} &= \mathbf{U}^T \mathbf{C} \mathbf{U} & \tilde{\mathbf{G}} &= \mathbf{U}^T \mathbf{G} \mathbf{U} \\ \tilde{\mathbf{B}} &= \mathbf{U}^T \mathbf{B} & \tilde{\mathbf{L}} &= \mathbf{U}^T \mathbf{L} \end{aligned} \quad (4.80)$$

and result in the same transfer matrix as if the Arnoldi based orthonormal basis were used:

$$\tilde{\mathbf{H}}(s) = \tilde{\mathbf{L}}^T (\tilde{\mathbf{G}} + s\tilde{\mathbf{C}})^{-1} \tilde{\mathbf{B}} \quad (4.81)$$

$$= \mathbf{L}^T \mathbf{U} (\mathbf{U}^T (\mathbf{G} + s\mathbf{C}) \mathbf{U})^{-1} \mathbf{U}^T \mathbf{B} \quad (4.82)$$

$$= \mathbf{L}^T \mathbf{X} \mathbf{Y} (\mathbf{Y}^T \mathbf{X}^T (\mathbf{G} + s\mathbf{C}) \mathbf{X} \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{X}^T \mathbf{B} \quad (4.83)$$

$$= \mathbf{L}^T \mathbf{X} (\mathbf{X}^T (\mathbf{G} + s\mathbf{C}) \mathbf{X})^{-1} \mathbf{X}^T \mathbf{B} \quad (4.84)$$

In [3, 13] also an optimal estimate for the Laguerre parameter α is given:

$$\alpha \simeq 2\pi f_{\max} \quad (4.85)$$

where f_{\max} is the system bandwidth. To give an intuitive explanation for this parameter, it must be noted that each function (4.74) can be interpreted as being a low-pass filter. Then it is straightforward to show that for $s = i\alpha$, where $i = \sqrt{-1}$, the 3dB-bandwidth of these filters are given. Therefore α is related to the maximum frequency of interest: f_{\max} .

Since, in the rest of this work, only the Laguerre-SVD technique will be used, the different steps in the algorithm will be repeated here [3]:

- select the values for q and α
- solve the equation $(\mathbf{G} + \alpha\mathbf{C})\mathbf{T}_0 = \mathbf{B}$ for \mathbf{T}_0
- solve the equation $(\mathbf{G} + \alpha\mathbf{C})\mathbf{T}_k = (\mathbf{G} - \alpha\mathbf{C})\mathbf{T}_{k-1}$, for $k = 1, 2, \dots, q-1$ making $\mathbf{T}_i = \mathbf{A}^i\mathbf{R}$
- construct the Krylov matrix $\mathbf{K} = [\mathbf{T}_0 \ \mathbf{T}_1 \ \dots \ \mathbf{T}_{q-1}]$
- calculate the singular value decomposition of \mathbf{K} : $\mathbf{K} = \mathbf{U}\Sigma\mathbf{V}^T$
- construct the matrices of the reduced order system

$$\begin{aligned}\tilde{\mathbf{C}} &= \mathbf{U}^T\mathbf{C}\mathbf{U} & \tilde{\mathbf{G}} &= \mathbf{U}^T\mathbf{G}\mathbf{U} \\ \tilde{\mathbf{B}} &= \mathbf{U}^T\mathbf{B} & \tilde{\mathbf{L}} &= \mathbf{U}^T\mathbf{L}\end{aligned}\quad (4.86)$$

For large systems, most time is consumed in the calculation of the LU-decomposition of $(\mathbf{G} + \alpha\mathbf{C})$ and in the calculation of the singular value decomposition of \mathbf{K} .

The parameter q will be called the *order of approximation*, and corresponds to the number of moments matched in (4.77). It also determines the size of the resulting matrices: $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{G}}$ are of dimension $pq \times pq$, $\tilde{\mathbf{B}}$ is of dimension $pq \times p$ and \mathbf{L} is of dimension $pq \times m$. It can be expected that, the higher q is chosen, the better the reduced model will approximate the original one, but at the same time the size and complexity of the reduced model increases.

4 FDTD AND ROM

4.1 Literature survey

Over the years a number of researchers have reported results where the FDTD method and a ROM algorithm were combined, very often this was the PVL algorithm. In almost every method the interest goes out to the approximate transfer matrix of an entire electromagnetic system, derived

from a state-space model, based on the spatially discretized problem space. Only in [18], a technique based on a subdomain approach is illustrated. However this technique was only verified in 1D.

The earliest papers [10–12, 19] used a slightly altered PVL algorithm to calculate the approximate transfer matrix, where the input was some problem specific source and where the output was some recorder. This implies that the problems that are involved have a limited number of ports, in [10] $m = p = 1$, and in [11, 12], the number of ports does not exceed two. A grid, as it is used in the FDTD method, is employed to generate a state-space model. In [10, 19] the grid was terminated by a perfect electric conducting material, in addition to this, in [11] and [12] the perfectly matched layer (PML) absorbing boundary condition was used to truncate the grid. As mentioned in Chapter 2, the use of perfect conducting materials to truncate the grid does preserve the diagonal structure of the \mathbf{C} matrix. Furthermore the inclusion of the PML material was performed in such a way that \mathbf{C} remained diagonal. This was then used advantageously by introducing the change of variables $\sigma = 1/s$:

$$\mathbf{h}(\sigma) = \frac{1}{\sigma} \mathbf{I}^T (\mathbf{I} + \sigma \mathbf{A})^{-1} \mathbf{r} \quad (4.87)$$

where the new matrices \mathbf{A} and \mathbf{r}

$$\mathbf{A} = -\mathbf{C}^{-1} \mathbf{G} \quad \mathbf{r} = \mathbf{C}^{-1} \mathbf{b} \quad (4.88)$$

have the same sparsity as the matrices \mathbf{G} and \mathbf{b} .

The PVL method was used to calculate the approximate transfer function of (4.87). Each step in the PVL method requires two matrix vector multiplications, corresponding to the calculation of $\mathbf{A}^i \mathbf{r}$ and $(\mathbf{A}^T)^i \mathbf{I}$. Since the matrix \mathbf{A} is as sparse as \mathbf{G} is, namely at most five non zero elements per row, these matrix vector multiplications are not very costly. As stressed in [11] and [12] the computational complexity of this matrix vector multiplication is equal to two standard FDTD updates. Furthermore in [10] and [19], where the grid was truncated by a perfect conducting material, this computational complexity could be reduced even further. Due to the skew symmetry of \mathbf{G} , the symmetric Lanczos process could be used. This amounted to a reduction by a factor of two. For each step a new moment is matched resulting in a better approximation. The approximated transfer function is then used, to obtain either a frequency result or a transient response, depending on the nature of the problem. This kind of approach does no longer involve time stepping.

The choice $\sigma = 1/s$, has as a consequence that the approximate model is valid for high s values. As a matter of fact the expansion point is chosen at infinity, $s_0 = \infty$. This corresponds to high frequencies or, equivalently

early times. This can be seen in the numerical examples presented in these papers. In [11] and [12], frequency results are shown, where the lowest frequency is 10 GHz. In [10], only transient pulses are shown, where, as the number of Lanczos iterations is increased, the original pulse response and the approximated pulse response coincide during a longer time. Since the sparsity of the original matrices is maintained when calculations with these matrices are involved, the size of the problems investigated, in number of variables of the original system (N), is quite high. E.g. in [10], $N = 189\,000$ is reported, and in [19] problems up to $N = 250\,000$ internal variables are investigated.

In [20], the regular PVL method, with expansion point $s_0 \neq 0$, is used to calculate the transfer function of a waveguide problem. As absorbing boundary conditions the perfectly matched layer is used and incorporated into the system. Since $s_0 \neq 0$, the matrix $(\mathbf{G} + s_0\mathbf{C})$ that needs to be inverted is no longer diagonal and

$$\mathbf{A} = -(\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C} \quad \mathbf{r} = (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{b} \quad (4.89)$$

are no longer sparse. It is not necessary to calculate $(\mathbf{G} + s_0\mathbf{C})^{-1}$ however, the LU decomposition of $(\mathbf{G} + s_0\mathbf{C})$ suffices. This is then used in a forward-backward substitution step to calculate $\mathbf{A}^{i+1}\mathbf{r}$ from $\mathbf{A}^i\mathbf{r}$. However this step is computationally more expensive than when (4.88) is used. This can also be noted by the size of the problems investigated: in [20] the largest problem contained 7665 unknowns, which is considerably less than the hundreds of thousands with $s_0 = \infty$. However, in this way, it is also possible to get an accurate solution at low frequencies.

For a special class of problems, namely transient diffusive electromagnetic fields, a technique based on PVL was devised in [21]. When large losses are involved, the term due to the time derivative is a lot smaller than the term due to the losses in Maxwell's second equation and the following approximation can be proposed:

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma_e \mathbf{E} \quad (4.90)$$

$$\simeq \sigma_e \mathbf{E} \quad (4.91)$$

Thanks to this, it is possible to use the symmetric Lanczos process, which requires only half as much work. The matrices \mathbf{A} and \mathbf{R} , of the Krylov subspace (4.33), are sparse, since here again the expansion point is chosen in this way that high frequency aspects are easily captured by the ROM.

Recently the idea surged to generate a reduced model of the original, FDTD based, state-space model, and instead of looking at the transfer function related to the reduced model, extract a circuit equivalent from

it [22]. These circuit equivalent models of general electromagnetic devices could then further be used in a circuit simulator. As ROM technique, the MPVL technique was used. The problem presented in [22], was a metallic T-shaped microstrip. The dimension of the original problem was $N = 496$.

All techniques discussed up to now did not have anything to do with the temporal part, namely the time stepping, of the FDTD algorithm. However, in [18] a technique is presented that combines a ROM technique with both the spatial and temporal part of the FDTD technique: a method is presented that generates a macromodel of a subdomain based on a fine grid and combines this ROM in the time domain with a coarse FDTD grid. The ROM technique used was the Efficient Nodal Order Reduction (ENOR) technique. This technique operates on transfer matrices of the form:

$$\mathbf{H}(s) = \mathbf{B}^T \left(s\mathbf{C} + \mathbf{G} + \frac{\mathbf{\Gamma}}{s} \right)^{-1} \mathbf{B} \quad (4.92)$$

and smaller matrices of the reduced system are obtained by orthogonal projection with respect to the basis \mathbf{V} . From the pole-residue representation of the transfer matrix, a set of time differential equations is derived. This set of equations is then discretized and used in combination with the surrounding FDTD grid to step in time. The numerical example in [18] was a 1D problem where the reflection of a plane wave due to a number of parallel dielectric layers was calculated. The refinement ratio between the fine grid used to derive the model and the coarse grid was $r = 3$. The time step that was used was always larger than the Courant limit related to the fine grid. It was reported that it was often possible to choose the time step of the coarse grid.

4.2 The subdomain FDTD method

The ROM step is, as stated in Chapter 3, essential before the time discretization, (3.26) and (3.28), of the subdomain model is performed. For each time step, the number of floating point operations per subdomain update, is in this way no longer $\mathcal{O}(N^2)$, but of the order $\mathcal{O}(p^2 q^2)$. By employing the real diagonalization, (3.34), the iteration matrix has at most two elements per row and an iteration is of the order $\mathcal{O}(p^2 q)$. The matrices linking the values of the input variables and the output variables to the values of the internal variables, i.e. from (3.34),

$$\Delta_t \mathbf{P}^{-1} \left(\tilde{\mathbf{C}} + \frac{\Delta_t}{2} \tilde{\mathbf{G}} \right)^{-1} \tilde{\mathbf{B}} \quad \text{and} \quad \tilde{\mathbf{L}}^T \mathbf{P} \quad (4.93)$$

where $(2\tilde{\mathbf{C}} + \Delta_t \tilde{\mathbf{G}})^{-1} (2\tilde{\mathbf{C}} - \Delta_t \tilde{\mathbf{G}}) = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^{-1}$, have no extra non-zero elements since they were dense, after the ROM algorithm was applied. This real di-

agonalization does not involve any loss of accuracy. The update, however, is still quadratical for p , whereas FDTD is linear to the number of field variables, so it is still important to keep p as small as possible. Keeping q as small as possible is also desired since we do not want to perform any unnecessary work.

First of all it is important to keep the order of approximation, q , as small as possible. However when q is chosen too small, the model will not be valid, or only in a small frequency region. It depends on the problem simulated how high the order of approximation q needs to be chosen.

Unfortunately, there is no easy way to determine the quality of the reduced model as a function of q . The best way to determine the frequency region for which the reduced model is a good approximation of the original system, is to try it out in a small simulation problem. Numerical experiments have shown that good models already become available for q values of two or three. The complexity of the subdomain determines the value of q . E.g. in [23] long 1D-like subdomains are used. For these subdomains, with high complexity, the order of approximation q had to be increased up to six to have a reasonable model of the subdomain. In other numerical examples [24], where small subgridded subdomains were reduced to function as a generalized FDTD subcell model, the complexity of the subdomain was low and good approximate models already exist for $q \leq 3$. These examples will be further discussed in Chapter 6.

The other parameter, that determines the size of the reduced system, is p , the number of input field variables of the subdomain. It is important to keep this number small, since each update is quadratical for this parameter. As a consequence, the subdomain algorithm is only beneficial when the subdomains are relatively small, or when the subdomain only communicates through a small number of variables with the surrounding grid. This is for example the case in [23], where perfect conductors are used to limit the number of input variables. There, only the field variables at the beginning and at the end of the subdomain determine the number of input variables, independently of the length of the subdomain.

This relation between p and computational complexity, clearly indicates the importance of subgridding. Subgridding allows to generate a complex model, with a high number of internal field variables, that can capture the high field variations, connected to the surrounding grid with only a small number of field variables.

Among all possible ROM techniques, the Laguerre-SVD technique was selected as a ROM technique. Although the Laguerre-SVD ROM technique is, due to the LU-decomposition of $(\mathbf{G} + \alpha\mathbf{C})$ and due to the SVD step, more laboursome than other ROM techniques, we propose to use it. There are

two reasons for this. First of all, the quality of the ROM's is very important and the Laguerre-SVD technique has the best cards: the orthonormal Laguerre functions used and the orthonormal basis for the Krylov matrix. Secondly, in contrast to most research trying to combine the FDTD method and ROM techniques, we are not interested in reducing the entire problem space, but only in reducing relatively small subdomains. The size of the original systems is only moderately high: several thousands as opposed to hundreds of thousands, making problems not prohibitively large for the Laguerre-SVD technique.

Hereafter, for two new FDTD algorithms, the different steps are layed down. The first algorithm is called the *subdomain FDTD algorithm* proposed in Section 3.2 of Chapter 3. The second algorithm is called the *generalized subdomain FDTD algorithm* proposed in Section 3.4 of Chapter 3. For each of these algorithms a number of steps need to be performed in advance; these steps need to be carried out only once. Once this preprocessing step has been done, the time stepping algorithm and the actual simulation can be started.

4.2.1 The subdomain FDTD algorithm

The subdomain FDTD algorithm is the combination of the standard FDTD equations and some time discretized reduced models derived from a subgridded subdomain. The final algorithm is illustrated using Fig. 4.1 In advance the following steps are performed:

1. Determine the areas with high field variations and apply subgridding there. In Fig. 4.1, this is inside cut C_1 .
2. Write out the state-space model for these subgridded subdomains C_1 .
3. Use a ROM technique, in our case this is the Laguerre-SVD technique, to generate a smaller reduced model.
4. If desired the reduced model is extended with the field variables located between C_1 and C_2 . Then a model of a subdomain belonging to C_2 is used. In Fig. 4.1, the subdomains belonging to C_1 and to C_2 are both O -type subdomains.
5. Discretize time for the subdomains, for O -type subdomains around $t = n\Delta_t$ and for P -type subdomains around $t = (n + 1/2)\Delta_t$.
6. Calculate the real eigendecomposition of the iteration matrix (3.31) and calculate the new matrices used in the update equation, e.g. for an O -type update equation (3.34).

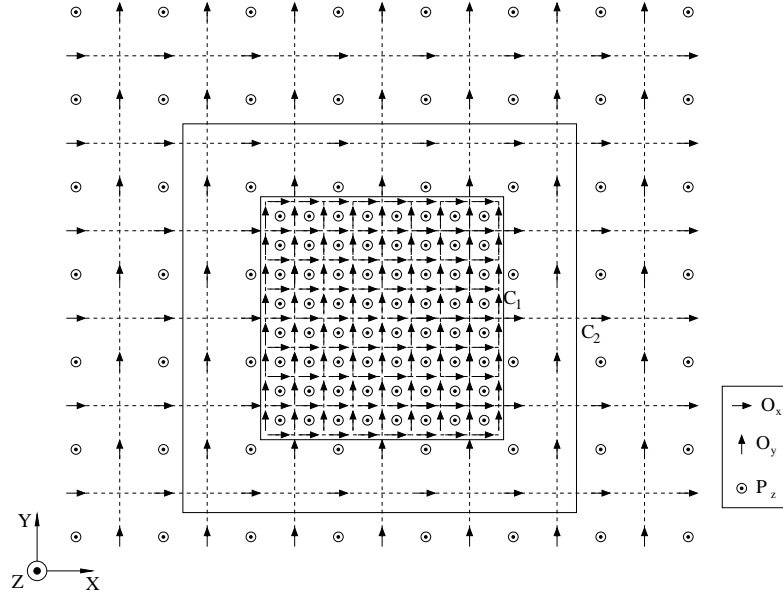


Figure 4.1: A sample of a typical subdomain FDTD grid, including one subdomain.

Once all this has been done, time stepping can be started.

1. Set $n = 0$.
2. Advance the O -type subdomains, the O_x field variables and the O_y field variables one time step.
3. Advance the P -type subdomains and the P_z field variables one time step.
4. Increment n and while $n < n_{\text{final}}$ go back to step 2.

The reason for extending the reduced subgridded subdomain, to obtain a model of the subdomain enclosed by C_2 , has everything to do with the time step for which stable results are obtained. It was found that the time step for which stable results are obtained, can be increased in this way. In any case, the time step is much larger than the Courant limit of the fine grid.

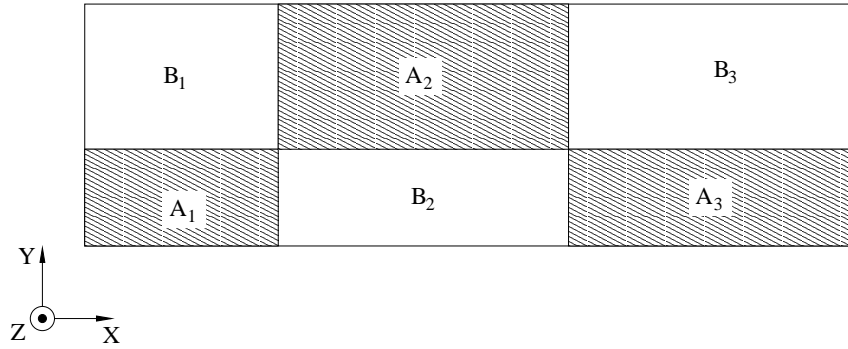


Figure 4.2: A typical grid used for the generalized subdomain FDTD algorithm. The two types of subdomains are indicated.

4.2.2 The generalized subdomain FDTD algorithm

The generalized subdomain FDTD algorithm can be seen as a higher level FDTD technique. It is illustrated by means of Fig. 4.2. In this figure no fields are shown since it is of no importance what field variables are at the boundary of a specific subdomain. In advance the following steps are performed:

1. Consider the problem space and subdivide it into a number of subdomains according to the rules explained in Section 3.4 of Chapter 3. In order to make the resulting process efficient, try to minimize, in each subdomain, the number of field variables that need to communicate with surrounding subdomains.
2. Divide the subdomains in two groups. In Fig. 4.2 the first group of subdomains carries a name containing the letter *A*, for the second group the names of the second group contain the letter *B*.
3. Write out the state-space model for each of these subdomains.
4. Generate a ROM for each of these systems.
5. Discretize time, for *A*-type subdomains this is around $t = n\Delta_t$ and for *B*-type subdomains this is around $t = (n + 1/2)\Delta_t$.
6. Calculate the real eigendecomposition of the iteration matrix (3.31) and calculate the new matrices used in the update equation, e.g. for an *A*-type update equation (3.34).

Once this has been done, time stepping can be started.

1. Set $n = 0$.
2. Advance the A -type subdomains one time step.
3. Advance the B -type subdomains one time step.
4. Increment n and while $n < n_{\text{final}}$ go back to step 2.

BIBLIOGRAPHY

- [1] R. W. Freund, "Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation," *Numer. Anal. Manuscript*, no. 98-3-02, Feb. 1998.
- [2] R. Achar and M. S. Nakhla, "Simulation of high-speed interconnects," *Proc. of the IEEE*, vol. 89, no. 5, pp. 693–728, May 2001.
- [3] L. Knockaert and D. De Zutter, "Passive reduced order multiport modeling: The Padé-Laguerre, Krylov-Arnoldi-SVD connection," *AEU Int. J. Electron Commun.*, vol. 53, no. 5, pp. 254–260, 1999.
- [4] P. Henrici, *Applied and Computational Complex Analysis, Vol. 1: Power Series, Integration, Conformal Mapping, Location of Zeros*, John Wiley & Sons, 1974.
- [5] D. L. Boley, "Krylov space methods on state-space control models," *Circuits Syst. Signal Processing*, vol. 13, no. 6, pp. 733–758, 1994.
- [6] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. on Computer-Aided Design*, vol. 14, no. 5, pp. 639–649, May 1995.
- [7] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, "A Lanczos-type method for multiple starting vectors," *Math. Comput.*, vol. 69, no. 232, pp. 1577–1601, Oct. 2000.
- [8] R. W. Freund, "Krylov-subspace methods for reduced-order modeling in circuit simulation," *J. Comput. Appl. Math.*, vol. 123, pp. 395–421, 2000.
- [9] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on Computer-Aided Design*, vol. 17, no. 8, pp. 645–654, Aug. 1998.

- [10] R. F. Remis and P. M. van den Berg, "A modified lanczos algorithm for the computation of transient electromagnetic wavefields," *IEEE Trans. Microwave Theory Tech.*, vol. 45, no. 12, pp. 2139-2149, Dec. 1997.
- [11] A. C. Cangellaris and L. Zhao, "Rapid FDTD simulation without time stepping," *IEEE Microwave and Guided Wave Letters*, vol. 9, no. 1, pp. 4-6, Jan. 1999.
- [12] A. C. Cangellaris, M. Celik, S. Pasha, and L. Zhao, "Electromagnetic model order reduction for system-level modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 6, pp. 840-850, Jun. 1999.
- [13] L. Knockaert and D. De Zutter, "Laguerre-SVD reduced-order modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1469-1475, Sep. 2000.
- [14] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability and model reduction," *IEEE Trans. Automatic Control*, vol. 26, no. 1, pp. 17-31, 1981.
- [15] L. Knockaert, B. Denecker, and D. De Zutter, "Explicitly reciprocal reduced order modeling: Laguerre-SVD versus balanced realizations," *IEEE Antennas Propagat. Symp.*, vol. 2, pp. 556-558, June 2002.
- [16] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on Computer-Aided Design*, vol. 9, no. 4, pp. 352-366, Apr. 1990.
- [17] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits and Systems*, vol. 22, no. 6, pp. 504-509, June 1975.
- [18] L. Kulas and M. Mrozowski, "Reduced-order models in FDTD," *IEEE Microwave and Wireless Comp. Letters*, vol. 11, no. 10, pp. 422-424, Oct. 2001.
- [19] A. C. Cangellaris and L. Zhao, "Rapid FDTD analysis of shielding enclosures," *IEEE Int. Symp on Electromagnetic Compatibility, Seattle*, vol. 2, pp. 827-831, 1999.
- [20] L. Zhao and A. C. Cangellaris, "Reduced-order modeling of electromagnetic field interactions in unbounded domains truncated by perfectly matched layers," *Microwave Opt. Techn. Letters*, vol. 17, no. 1, pp. 62-66, Jan. 1998.

- [21] R. F. Remis and P. M. van den Berg, "Efficient computation of transient diffusive electromagnetic fields by a reduced modeling technique," *Radio Science*, vol. 33, no. 2, pp. 191–204, Mar. 1998.
- [22] I. Munteanu, T. Wittig, T. Weiland, and D. Ioan, "FIT/PVL circuit-parameter extraction for general electromagnetic devices," *IEEE Trans. Magnetics*, vol. 36, no. 4, pp. 1421–1425, July 2000.
- [23] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," *IEEE Microwave and Guided Wave Letters*, vol. 10, no. 8, pp. 301–303, Aug. 2000.
- [24] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Generation of FDTD subcell equations by means of reduced order modeling," *IEEE Trans. Antennas Propagat.*, accepted for publication.

CHAPTER 5

Stability

1 INTRODUCTION

An important issue related to the finite difference time domain method is stability. Since the FDTD method is an explicit iteration scheme, it is vital to know when results will suffer from instability. An example of a result where instability has polluted the signal is shown in Fig. 5.1. The example should show an incident pulse and a reflected pulse. In the figure the incident pulse can easily be distinguished. The reflected pulse, however, is infected by an ever-growing, spurious signal. This spurious signal increases without limit as time goes on. This kind of problem is also referred to as *late-time instability*.

Since the FDTD equations formulate a fully *explicit* marching-in-time method, i.e. the update of a field variable only depends on values of field variables at previous time steps, a condition for stability can be formulated. This stability condition for the FDTD method is known as the Courant condition, or is sometimes also referred to as the Courant-Friedrich-Levy (CFL) condition [1]. It imposes a maximum value for the time step. This maximum time step is related to the space step used in the grid.

Not all FDTD methods are explicit: the recently introduced ADI-FDTD method [2], the Alternating Direction Implicit FDTD method, is an *implicit* method. The update of a field variable does not only involve values of field variables at previous time steps, but it also involves the values of field variables at the current time step. Implicit time stepping algorithms involve matrix inversions, making them more labour intensive. Fortunately, in the ADI-FDTD method, this matrix inversion can be simplified, as was explained in Section 4.2 of Chapter 3. The advantage of this implicit method is that the time step is no longer restricted, which means that the method

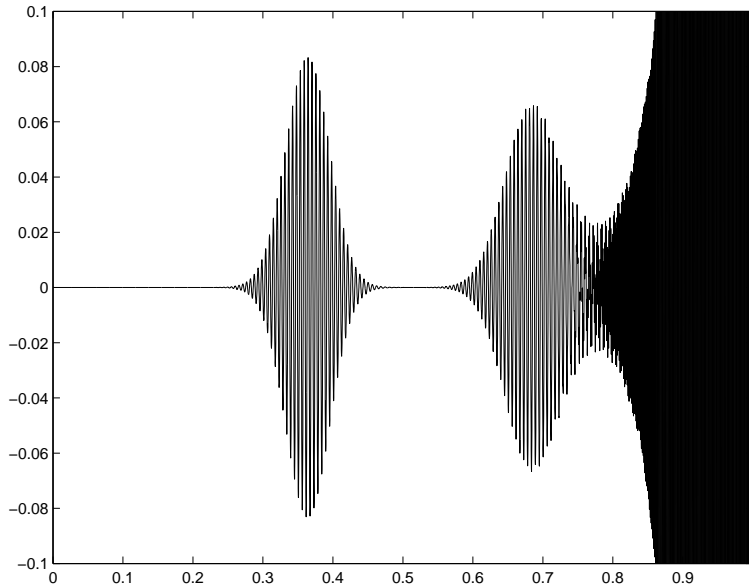


Figure 5.1: A typical result where the useful signal is drowned out by the instability.

is unconditionally stable, see [2] for 2D and [3] for 3D.

We will not repeat the classical stability derivations as they also appear in [4]. There the numerical stability is investigated with the Fourier method, also called the Von Neumann method. The lattice numerical plane waves that propagate are Fourier transformed, resulting in a spatial spectral domain representation of the system. Then stability is ensured by checking if the magnitude of these spatial Fourier components remains bounded as a function of time. If this is the case, the scheme is stable, if not, the scheme is potentially unstable. The stability conditions derived in this way apply to infinite grids where the medium is invariant, either vacuum or a lossless isotropic medium. Together with the stability condition, the numerical dispersion related to the grid and the time step can be analyzed.

In contrast to this approach, stability will be investigated here, starting from the FDTD equations in matrix form. Using linear algebra and further not relying on the physics of the problem involved, some considerations concerning stability can be derived. By using this approach, the stability condition for truncated grids can be derived, which is similar to the known Courant condition. It will also enable us to say something about the stability condition for the subdomain FDTD scheme.

As it also has become apparent in the previous chapters of this work, we clearly distinguish the spatial and the temporal aspects of the FDTD method. This will become clear once more in this chapter: from this several reasons causing instability can be distinguished. In Section 2, the effect of the spatial discretization will be considered. The poles of the state-space system, associated with the discretized grid, have to be analyzed and have to be in the left half plane. In Section 3, the reduced order modeling algorithm will be investigated. More specifically, the effect it has on the poles of the original state-space system will be discussed. In Section 4, the effect of discretizing time is under discussion. Three different time discretization schemes will be investigated: the standard FDTD equations, the ADI-FDTD method, the new subdomain algorithm. As we go along, it will become clear that for each step, something can go wrong causing the final algorithm to suffer from instability. When some cause of instability was introduced in the spatial discretization step, it is unlikely that this can be removed in further steps, i.e. ROM or time discretization. This instability will sometimes be very difficult to avoid while maintaining other properties of the algorithm, especially the relation between accurate subgridding and stability is very difficult.

2 SPATIAL DISCRETIZATION RELATED STABILITY

2.1 Spatially reciprocal grids

2.1.1 The lossless case

As explained in Chapter 2, for a uniform orthogonal grid, the set of first order differential equations can be written as follows:

$$\begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_\sigma & \frac{c_0}{\Delta} \mathbf{K} \\ -\frac{c_0}{\Delta} \mathbf{K}^T & \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (5.1)$$

The matrices \mathbf{D}_θ , \mathbf{D}_κ are diagonal and positive definite since all elements are strictly positive real numbers. The matrices \mathbf{D}_σ , \mathbf{D}_{σ^*} are diagonal and positive semidefinite since all elements are zero or positive real numbers. This kind of grid maintains the spatial reciprocity: when in the equation of a field variable, e.g. $O_x|_{i+1/2,j}$, a neighbouring field variable is present, e.g. $P_z|_{i+1/2,j+1/2}$ is present, then, reversely, in the equation of the field variable $P_z|_{i+1/2,j+1/2}$, the field variable $O_x|_{i+1/2,j}$ is present. The coefficients are equal

but opposite in sign. This can then be seen in the state-space model (5.1): $\mathbf{G}_{12}^T = -\mathbf{G}_{21}$, when the state-space model is $\mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x}$ and $\mathbf{x}^T = \begin{bmatrix} \mathbf{o}^T & \mathbf{p}^T \end{bmatrix}$.

When a lossless medium is concerned, $\mathbf{D}_\sigma = 0$ and $\mathbf{D}_{\sigma^*} = 0$, equation (5.1) becomes:

$$\begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} 0 & \frac{c_0}{\Delta} \mathbf{K} \\ -\frac{c_0}{\Delta} \mathbf{K}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (5.2)$$

and the matrix \mathbf{G} becomes skew symmetric, $\mathbf{G} = -\mathbf{G}^T$. The matrix \mathbf{C} is diagonal and since all diagonal elements are strict positive real numbers, the square root of this matrix can be introduced:

$$\mathbf{C} = \mathbf{C}^{1/2} \mathbf{C}^{1/2} \quad (5.3)$$

where $\mathbf{C}^{1/2}$ is diagonal:

$$[\mathbf{C}^{1/2}]_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ \sqrt{[\mathbf{C}]_{ii}} & \text{for } i = j \end{cases} \quad (5.4)$$

By changing the variables to

$$\mathbf{x}' = \begin{bmatrix} \mathbf{o}' \\ \mathbf{p}' \end{bmatrix} = \begin{bmatrix} \mathbf{D}_\theta^{1/2} \mathbf{o} \\ \mathbf{D}_\kappa^{1/2} \mathbf{p} \end{bmatrix} = \mathbf{C}^{1/2} \mathbf{x} \quad (5.5)$$

and by left multiplying with $\mathbf{C}^{-1/2}$, the system (5.2) becomes:

$$\dot{\mathbf{x}}' = -\mathbf{C}^{-1/2} \mathbf{G} \mathbf{C}^{-1/2} \mathbf{x}' = -\mathbf{A} \mathbf{x}' \quad (5.6)$$

The matrix \mathbf{A}

$$\mathbf{A} = \mathbf{C}^{-1/2} \mathbf{G} \mathbf{C}^{-1/2} = \begin{bmatrix} 0 & \frac{c_0}{\Delta} \mathbf{D}_\theta^{-1/2} \mathbf{K} \mathbf{D}_\kappa^{-1/2} \\ -\frac{c_0}{\Delta} \mathbf{D}_\kappa^{-1/2} \mathbf{K}^T \mathbf{D}_\theta^{-1/2} & 0 \end{bmatrix} \quad (5.7)$$

is still skew symmetric. The poles of the system (5.2), that determine the stability of the system, are the negative eigenvalues $(-\lambda)$ of \mathbf{A} . The change of variables does not change this.

Since the matrix \mathbf{A} is real and skew symmetric, $\mathbf{A}^T = -\mathbf{A}$, it is easy to show that all eigenvalues of \mathbf{A} and hence all poles are purely imaginary numbers. Suppose λ_i is an eigenvalue of \mathbf{A} and \mathbf{x}_i is the corresponding eigenvector:

$$\mathbf{A} \mathbf{x}_i = \lambda_i \mathbf{x}_i \quad (5.8)$$

By right multiplying this with the Hermitian transpose of \mathbf{x}_i , $\mathbf{x}_i^H = (\mathbf{x}_i^*)^T$, with $*$ denoting complex conjugate, we get:

$$\mathbf{x}_i^H \mathbf{A} \mathbf{x}_i = \lambda_i \mathbf{x}_i^H \mathbf{x}_i \quad (5.9)$$

Taking the self-adjoint of the left hand side

$$(\mathbf{x}_i^H \mathbf{A} \mathbf{x}_i)^H = \mathbf{x}_i^H \mathbf{A}^H \mathbf{x}_i \quad (5.10)$$

$$= -\mathbf{x}_i^H \mathbf{A} \mathbf{x}_i \quad (5.11)$$

$$= -\lambda_i \mathbf{x}_i^H \mathbf{x}_i \quad (5.12)$$

and of the right hand side

$$(\lambda_i \mathbf{x}_i^H \mathbf{x}_i)^H = \lambda_i^* \mathbf{x}_i^H \mathbf{x}_i \quad (5.13)$$

the following equation is obtained

$$-\lambda_i \mathbf{x}_i^H \mathbf{x}_i = \lambda_i^* \mathbf{x}_i^H \mathbf{x}_i \quad (5.14)$$

Since, by definition, for an eigenvector \mathbf{x}_i , $\mathbf{x}_i^H \mathbf{x}_i \neq 0$, the relation $\lambda_i^* = -\lambda_i$ holds for each eigenvalue. This is only possible when $\text{Re}[\lambda_i] = 0$ and all eigenvalues of \mathbf{A} are located on the imaginary axis. Since all eigenvalues are on the imaginary axis, no poles are in the right half plane. The question whether all poles on the imaginary axis are single, required to be able to speak of a stable system, has to be answered in a non-affirmative way: at least $|\dim[\mathbf{o}] - \dim[\mathbf{p}]|$ poles are located in the origin. But since there is still another step in the process, the time discretization step, we will only care about poles in the right half plane, because time discretization changes the criteria for stability.

For each skew symmetric matrix \mathbf{A} , the eigenvalues are imaginary. Hence, skew symmetry of \mathbf{G} , and automatically related to it, reciprocity of the spatial discretization, is important. If one is able to show, by changing the variables, that a system is similar to a system that can be described by a skew symmetric matrix, then that system is stable.

2.1.2 The lossy case

When the lossy case is considered, and by changing the variables as in (5.5), a similar system matrix is obtained:

$$\mathbf{A} = \begin{bmatrix} \mathbf{D}_\theta^{-1/2} \mathbf{D}_\sigma \mathbf{D}_\theta^{-1/2} & \frac{c_0}{\Delta} \mathbf{D}_\theta^{-1/2} \mathbf{K} \mathbf{D}_\kappa^{-1/2} \\ -\frac{c_0}{\Delta} \mathbf{D}_\kappa^{-1/2} \mathbf{K}^T \mathbf{D}_\theta^{-1/2} & \mathbf{D}_\kappa^{-1/2} \mathbf{D}_{\sigma^*} \mathbf{D}_\kappa^{-1/2} \end{bmatrix} \quad (5.15)$$

The eigenvalues associated to (5.15) are no longer imaginary. It is possible to indicate the region in the complex domain where all the eigenvalues are located. For that we need to introduce the symmetric and skew sym-

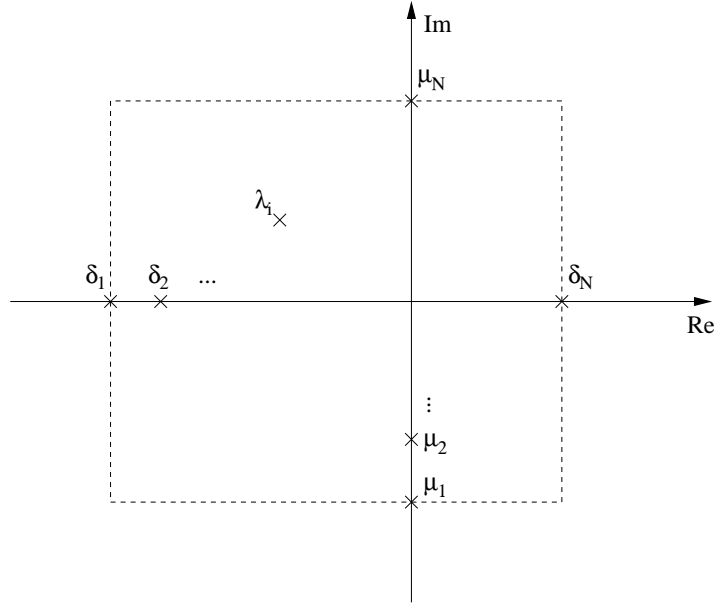


Figure 5.2: The eigenvalues associated to a real matrix \mathbf{A} , are located inside or on the boundary of the rectangle shown.

metric part of a matrix:

$$\mathbf{S} = \frac{\mathbf{A} + \mathbf{A}^T}{2} = \mathbf{S}^T \quad (5.16a)$$

$$\mathbf{R} = \frac{\mathbf{A} - \mathbf{A}^T}{2} = -\mathbf{R}^T \quad (5.16b)$$

where \mathbf{S} is the symmetric part and \mathbf{R} is the skew symmetric part of the real matrix \mathbf{A} . Note first of all that the eigenvalues of a symmetric matrix are real numbers and the eigenvalues of a skew symmetric matrix are, as shown above, imaginary numbers. When the eigenvalues of \mathbf{S} are δ_i for $i = 1, 2, \dots, N$, where $\delta_1 \leq \delta_2 \leq \dots \leq \delta_N$ and when the eigenvalues of \mathbf{R} are $j\mu_i$ for $i = 1, 2, \dots, N$, where $\mu_1 \leq \mu_2 \leq \dots \leq \mu_N$ and where $j = \sqrt{-1}$. Then, see [5], for each eigenvalue λ_i of \mathbf{A} :

$$\delta_1 \leq \text{Re}(\lambda_i) \leq \delta_N \quad \text{for } i = 1, 2, \dots, N \quad (5.17)$$

$$\mu_1 \leq \text{Im}(\lambda_i) \leq \mu_N \quad \text{for } i = 1, 2, \dots, N \quad (5.18)$$

In other words each eigenvalue of \mathbf{A} is located in the rectangle shown in Fig 5.2.

Applying this knowledge to the matrix (5.15), we can separate the matrix as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{D}_\theta^{-1/2} \mathbf{D}_\sigma \mathbf{D}_\theta^{-1/2} & 0 \\ 0 & \mathbf{D}_\kappa^{-1/2} \mathbf{D}_{\sigma^*} \mathbf{D}_\kappa^{-1/2} \end{bmatrix} \quad (5.19)$$

$$\mathbf{R} = \begin{bmatrix} 0 & \frac{c_0}{\Delta} \mathbf{D}_\theta^{-1/2} \mathbf{K} \mathbf{D}_\kappa^{-1/2} \\ -\frac{c_0}{\Delta} \mathbf{D}_\kappa^{-1/2} \mathbf{K}^T \mathbf{D}_\theta^{-1/2} & 0 \end{bmatrix} \quad (5.20)$$

since the diagonal blocks of \mathbf{A} are symmetric, and the off-diagonal blocks of \mathbf{A} form the skew symmetric part. We are especially interested in the width of the rectangle, or in other words in the eigenvalues of \mathbf{S} . This matrix determines whether the poles are possibly located in the right half plane. The matrix \mathbf{S} is diagonal, and hence the eigenvalues are equal to the elements on the diagonal. All these elements have the following value: a loss factor divided by a relative constitutive parameter associated to the field variable. Since both are positive, it can be concluded that each eigenvalue of \mathbf{A} has a positive real part. Then as a consequence each pole has a negative real part, or no poles are located in the right half plane.

It is possible to generalize this. A system described by:

$$\begin{bmatrix} \mathbf{C}_{11} & 0 \\ 0 & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (5.21)$$

where \mathbf{C}_{11} , \mathbf{C}_{22} are diagonal, positive definite and \mathbf{G}_{11} , \mathbf{G}_{22} are diagonal positive semi-definite and where

$$-\mathbf{D}_1 \mathbf{G}_{12} \mathbf{D}_2 = \mathbf{G}_{21}^T \quad (5.22)$$

can be expressed by a single system matrix:

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = - \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ -\mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (5.23)$$

when \mathbf{D}_1 and \mathbf{D}_2 are diagonal positive definite. This implies that \mathbf{G}_{12} and \mathbf{G}_{21}^T have the same sparsity but need not be opposite. The stability is then determined by the positive semi-definiteness of \mathbf{A}_{11} and \mathbf{A}_{22} . To show this, first write (5.21) as:

$$\begin{aligned} & \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{D}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{11} & 0 \\ 0 & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1^{-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} \\ &= - \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{D}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ -\mathbf{D}_2 \mathbf{G}_{12}^T \mathbf{D}_1 & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1^{-1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \end{aligned} \quad (5.24)$$

where the inverse of \mathbf{D}_1 and \mathbf{D}_2 exist since they are assumed to be positive definite. This equation can be compactly written as:

$$\begin{bmatrix} \mathbf{C}_{11}\mathbf{D}_1^{-1} & 0 \\ 0 & \mathbf{D}_2^{-1}\mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1\dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{G}_{11}\mathbf{D}_1^{-1} & \mathbf{G}_{12} \\ -\mathbf{G}_{12}^T & \mathbf{D}_2^{-1}\mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1\mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (5.25)$$

In a similar fashion as before, taking the square root of the diagonal blocks of the new \mathbf{C} matrix, and since operations on diagonal matrices are similar to operations on real numbers, more specifically commutativity of the multiplication and the square root are assured, this can easily be transformed to

$$\begin{bmatrix} \mathbf{C}_{11}^{1/2}\mathbf{D}_1^{1/2}\dot{\mathbf{o}} \\ \mathbf{C}_{22}^{1/2}\mathbf{D}_2^{-1/2}\dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{C}_{11}^{-1}\mathbf{G}_{11} & \mathbf{C}_{11}^{-1/2}\mathbf{D}_1^{1/2}\mathbf{G}_{12}\mathbf{D}_2^{1/2}\mathbf{C}_{22}^{-1/2} \\ -\mathbf{C}_{22}^{-1/2}\mathbf{D}_2^{1/2}\mathbf{G}_{12}^T\mathbf{D}_1^{1/2}\mathbf{C}_{11}^{-1/2} & \mathbf{C}_{22}^{-1}\mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{11}^{1/2}\mathbf{D}_1^{1/2}\mathbf{o} \\ \mathbf{C}_{22}^{1/2}\mathbf{D}_2^{-1/2}\mathbf{p} \end{bmatrix} \quad (5.26)$$

This result can be used to prove that certain subcell models do not disturb the stability of the system. Take, for example, when working in the TM-case, the subcell model for a thin perfectly conducting wire [6]. In Fig. 5.3, a part of a simulation problem is shown incorporating a thin wire with radius $r_0 < \Delta$. The wire is located at $((i + 1/2)\Delta, (j + 1/2)\Delta)$. This small feature would, in the standard FDTD method, require a very fine discretization. With a subcell model, this can be overcome. A $1/r$ behaviour, where r is the distance from the wire center, for the circumferential magnetic field is assumed. Based on this local behaviour and using Faraday's law in integral form, a new equation for the fields $O_x|_{i+1/2,j}$, $O_x|_{i+1/2,j+1}$, $O_y|_{i,j+1/2}$ and $O_y|_{i+1,j+1/2}$, shown in Fig. 5.3, is derived. For the field variable $O_y|_{i+1,j+1/2}$, this is

$$\frac{dO_y|_{i+1,j+1/2}}{dt} = \frac{c_0}{\frac{\Delta}{2} \ln\left(\frac{\Delta}{r_0}\right)} P_z|_{i+3/2,j+1/2} \quad (5.27)$$

The field variable $P_z|_{i+1/2,j+1/2}$ is zero.

For this subcell model, the choice for \mathbf{D}_1 and \mathbf{D}_2 is straightforward. Choose $\mathbf{D}_2 = \mathbf{I}$, and

$$[\mathbf{D}_1]_{ii} = \begin{cases} \frac{\ln\left(\frac{\Delta}{r_0}\right)}{2} & \text{for } i = i_1, i_2, i_3, i_4 \\ 1 & \text{else} \end{cases} \quad (5.28)$$

where for the vector \mathbf{o} , i_1 is the index of $O_x|_{i+1/2,j}$, i_2 the index of $O_x|_{i+1/2,j+1}$, i_3 the index of $O_y|_{i,j+1/2}$ and i_4 the index of $O_y|_{i+1,j+1/2}$.

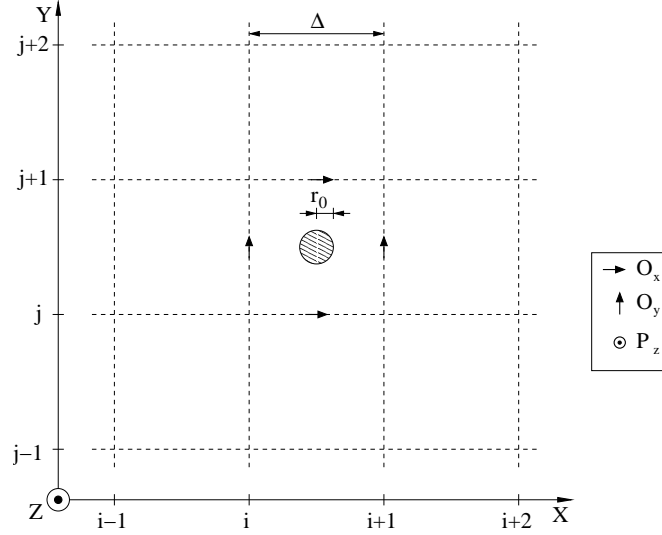


Figure 5.3: A thin wire located at $((i + \frac{1}{2})\Delta, (j + \frac{1}{2})\Delta)$ can be modeled by changing the equations for the four surrounding field variables indicated in the figure.

2.2 Spatially non-reciprocal grids

2.2.1 The lossless case

As has been stressed in the previous section, the spatial reciprocity is a property strongly related to the stability. Subgridding is an example of a technique developed to enhance the capabilities of the FDTD method. Subgridding has difficulties in maintaining this spatial reciprocity: only a small number of papers reported spatial discretization schemes where the spatial reciprocity was maintained [7], [8], [9], [10]. Others have also reported a spatially reciprocal contour path FDTD method (CPFDTD) [11], resulting in a stable algorithm. The consequences of the loss of spatial reciprocity, in lossless media, will be investigated here.

The system matrix, describing a lossless, spatially discretized medium, where subgridding was applied in certain regions and truncated by a perfect conducting material looks like:

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{A}_{12} \\ \mathbf{A}_{21} & 0 \end{bmatrix} \quad (5.29)$$

As explained in Section 3 of Chapter 2, in general $\mathbf{A}_{12} \neq -\mathbf{A}_{21}$, this is especially the case when accuracy is a major concern. By writing \mathbf{A} as the

sum of a symmetric, \mathbf{S} , and a skew symmetric, \mathbf{R} , matrix (5.16), we can simplify the problem to determining the smallest eigenvalue of

$$\mathbf{S} = \begin{bmatrix} 0 & \frac{1}{2}(\mathbf{A}_{12} + \mathbf{A}_{21}^T) \\ \frac{1}{2}(\mathbf{A}_{12}^T + \mathbf{A}_{21}) & 0 \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{S}_{12} \\ \mathbf{S}_{12}^T & 0 \end{bmatrix} \quad (5.30)$$

If some eigenvalue of \mathbf{S} is negative, then the possibility of poles in the right half plane exists.

To find out something about the eigenvalues of \mathbf{S} , we need to introduce the 'large' singular value decomposition of \mathbf{S}_{12} [12]:

$$\mathbf{S}_{12} = \mathbf{U}^T \Sigma \mathbf{V} \quad (5.31)$$

where, for $n_1 = \dim[\mathbf{o}]$ and $n_2 = \dim[\mathbf{p}]$, $\mathbf{S}_{12}, \Sigma \in \mathbb{R}^{n_1 \times n_2}$, $\mathbf{U} \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times n_2}$. It has to be noted that in our case $n_1 \geq n_2$, since the number of O_x and O_y field variables is larger than the number P_z field variables. The matrices \mathbf{U} and \mathbf{V} are orthogonal:

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I} \quad (5.32a)$$

$$\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I} \quad (5.32b)$$

whereas, in the 'small' singular value decomposition $\Sigma \in \mathbb{R}^{n_2 \times n_2}$, $\mathbf{U} \in \mathbb{R}^{n_1 \times n_2}$ is only column orthogonal, $\mathbf{U}^T \mathbf{U} = \mathbf{I}$. The matrix Σ is diagonal:

$$[\Sigma]_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ \sigma_i & \text{for } i = j \end{cases} \quad (5.33)$$

and the diagonal elements are the singular values of \mathbf{S}_{12} . The matrix \mathbf{S} can now be rewritten as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{U}^T & 0 \\ 0 & \mathbf{V}^T \end{bmatrix} \begin{bmatrix} 0 & \Sigma \\ \Sigma^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{V} \end{bmatrix} \quad (5.34)$$

By introducing the permutation matrix \mathbf{P} , which has exactly one 1 in each row and each column and where its transpose is the inverse, $\mathbf{P}^T \mathbf{P} = \mathbf{I}$:

$$[\mathbf{P}]_{ij} = \begin{cases} 1 & \text{for } i = 2k - 1, j = k \quad k = 1, 2, \dots, n_2 \\ 1 & \text{for } i = 2k, j = n_1 + k \quad k = 1, 2, \dots, n_2 \\ 1 & \text{for } i = 2n_2 + k, j = n_2 + k \quad k = 1, 2, \dots, n_1 - n_2 \\ 0 & \text{else} \end{cases} \quad (5.35)$$

it is possible to block diagonalize, with blocks of size one and two, the matrix \mathbf{A} :

$$\mathbf{S} = \begin{bmatrix} \mathbf{U}^T & 0 \\ 0 & \mathbf{V}^T \end{bmatrix} \mathbf{P} \Omega \mathbf{P}^T \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{V} \end{bmatrix} \quad (5.36)$$

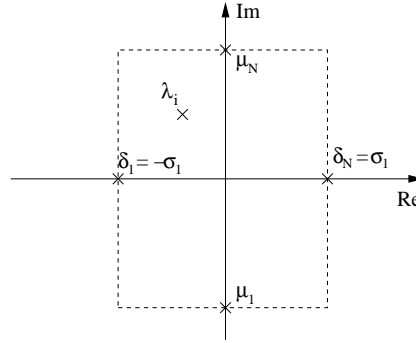


Figure 5.4: The eigenvalues associated to a real matrix \mathbf{A} , with $\mathbf{A}_{11} = 0$ and $\mathbf{A}_{22} = 0$ and arbitrary \mathbf{A}_{12} and \mathbf{A}_{21} are located inside or on the boundary of the dashed rectangle.

where Ω is

$$\Omega = \text{diag} \left\{ \begin{bmatrix} 0 & \sigma_1 \\ \sigma_1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & \sigma_{n_2} \\ \sigma_{n_2} & 0 \end{bmatrix}, 0, \dots, 0 \right\} \quad (5.37)$$

a block diagonal matrix, with n_2 leading 2×2 blocks, and $n_1 - n_2$ zeros. The eigenvalues of \mathbf{S} are the eigenvalues of Ω ; this corresponds to the eigenvalues of the different blocks in Ω . It can be seen that the eigenvalues appear in pairs: when $\lambda = \sigma_i$ is an eigenvalue, then $\lambda = -\sigma_i$ is also an eigenvalue. When the singular values of \mathbf{S}_{12} are ordered $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_2}$, then $\delta_1 = -\sigma_1$ and $\delta_N = \sigma_1$. This means that we are not able to demarcate an area for the eigenvalues of \mathbf{A} , in such a way that the corresponding poles, being the negatives of the eigenvalues of \mathbf{A} , are not in the right half plane. In Fig. 5.4, this area has been shown and one half of it is in the left half plane, the other half is in the right half plane. In this way, it is not possible to assure the stability of the system matrix.

However, the bounds set by \mathbf{S} are exaggerated. It is a sufficient condition, but not a necessary condition. To get an idea whether, for the subgridding case, unstable poles are present, a small example will be investigated. In Fig. 5.5, a small problem domain has been shown. The grid is truncated by a perfect conducting material: $P_z = 0$ at $x = -\frac{3}{2}\Delta_f$, $x = 18 + \frac{3}{2}\Delta_f$, $y = -\frac{3}{2}\Delta_f$ and $y = 18 + \frac{3}{2}\Delta_f$. For the TM case, this would mean a problem space truncated by a perfect electric conductor. The grid covers an area of 6×6 coarse cells, and in the middle region, covering 2×2 coarse cells, subgridding has been applied with a refinement ratio $r = 3$. The field variables can be divided into 4 groups:

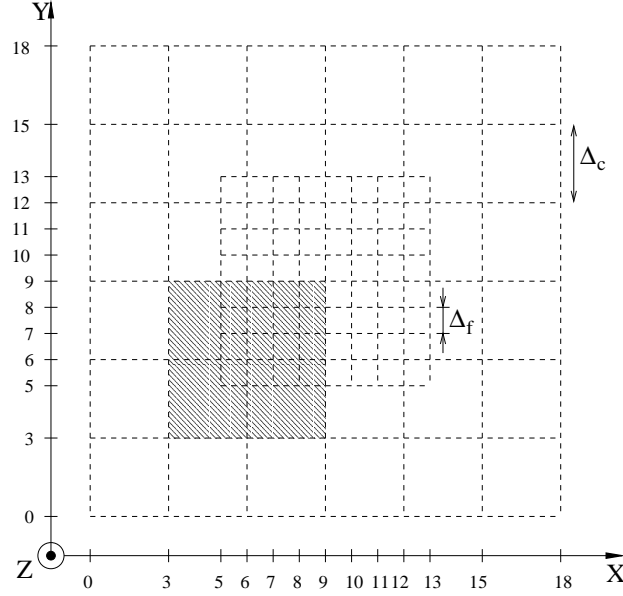


Figure 5.5: A small grid covering an area of 6 by 6 coarse cells, where, in a region covering 2 by 2 coarse cells, subgridding ($r = 3$) has been applied.

- 144 O -type fine grid field variables: O_x and O_y field variables where $x \in [5\Delta_f, 13\Delta_f]$ and $y \in [5\Delta_f, 13\Delta_f]$
- 72 O -type coarse grid field variables: O_x and O_y field variables where either $x \in [0, \frac{9}{2}\Delta_f]$, $x \in [(13+\frac{1}{2})\Delta_f, 18\Delta_f]$, $y \in [0, \frac{9}{2}\Delta_f]$ or $y \in [(13+\frac{1}{2})\Delta_f, 18\Delta_f]$
- 64 P -type fine grid field variables: P_z field variables where both $x \in [5\Delta_f, 13\Delta_f]$ and $y \in [5\Delta_f, 13\Delta_f]$
- 32 P -type coarse grid field variables: O_x and O_y field variables where either $x \in [0, \frac{9}{2}\Delta_f]$, $x \in [(13+\frac{1}{2})\Delta_f, 18\Delta_f]$, $y \in [0, \frac{9}{2}\Delta_f]$ or $y \in [(13+\frac{1}{2})\Delta_f, 18\Delta_f]$

In an effort not to overload the figure, only the Yee cells and not the field variables have been shown. The fine grid boundary field variables use the simple linear interpolation, e.g.:

$$\frac{\partial P_z|_{11/2,5}}{\partial y} \simeq \frac{P_z|_{11/2,11/2} - \frac{2}{3} P_z|_{9/2,9/2} - \frac{1}{3} P_z|_{15/2,9/2}}{\Delta} \quad (5.38)$$

Instead of calculating the eigenvalues of the entire system matrix (5.29), we first note that, for an eigenvalue λ of \mathbf{A} and corresponding eigenvector \mathbf{x} :

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 0 & \mathbf{A}_{12} \\ \mathbf{A}_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (5.39)$$

Writing this out for the different blocks

$$\begin{cases} \mathbf{A}_{12}\mathbf{x}_2 = \lambda\mathbf{x}_1 \\ \mathbf{A}_{21}\mathbf{x}_1 = \lambda\mathbf{x}_2 \end{cases} \quad (5.40)$$

and right multiplying the first equation of (5.40) with \mathbf{A}_{21} and using the second equation of (5.40) gives

$$\mathbf{A}_{21}\mathbf{A}_{12}\mathbf{x}_2 = \lambda\mathbf{A}_{21}\mathbf{x}_1 \quad (5.41)$$

$$= \lambda^2\mathbf{x}_2 \quad (5.42)$$

Therefore, it suffices to determine the eigenvalues of the product of the off-diagonal blocks of the original system matrix, since these eigenvalues correspond to the square of the eigenvalues of \mathbf{A} . Since $\mathbf{A}_{21}\mathbf{A}_{12}$ is a real matrix, its eigenvalues are either real or come in complex conjugate pairs. When the product $\mathbf{A}_{21}\mathbf{A}_{12}$ has a pair of complex conjugate eigenvalues, then the matrix \mathbf{A} has at least two eigenvalues in the right half plane. Hence, the eigenvalues of $\mathbf{A}_{21}\mathbf{A}_{12}$ need to be real.

It is possible to perform the same calculation for $\mathbf{A}_{12}\mathbf{A}_{21} \in \mathbb{R}^{n_1 \times n_1}$. This then leads to the same eigenvalues and conclusions. However, since $n_2 \leq n_1$, and $\mathbf{A}_{21}\mathbf{A}_{12} \in \mathbb{R}^{n_2 \times n_2}$, it is numerically more interesting to calculate the eigenvalues for the smallest matrix. The eigenvalues of $\mathbf{A}_{12}\mathbf{A}_{21}$ are the eigenvalues of $\mathbf{A}_{21}\mathbf{A}_{12}$ but with $n_1 - n_2$ zero eigenvalues extra. In the example of Fig. 5.5, $n_1 = 216$ and $n_2 = 96$.

As a first step we consider the medium to be vacuum filled. The corresponding eigenvalues of $\mathbf{A}_{21}\mathbf{A}_{12}$, were all strictly negative:

$$-7.759 \left(\frac{c_0}{\Delta_f} \right)^2 \leq \lambda^2 \leq -0.0444 \left(\frac{c_0}{\Delta_f} \right)^2 \quad (5.43)$$

The eigenvalues are shown in Fig. 5.6, and are indicated with circles (\circ). This shows that, when $\mathbf{A}_{12} \neq -\mathbf{A}_{21}^T$, the problem space will not necessarily correspond to an unstable problem. Therefore, the spatial reciprocity property is a sufficient but not a necessary condition for stability.

As a second step an isotropic material, with $\kappa_r \neq 1$, was added. This is the shaded region in Fig 5.5, and the value was: $\kappa_r = 16$. For a TM-problem, this material is a dielectric with $\epsilon_r = 16$. The material crosses the fine grid

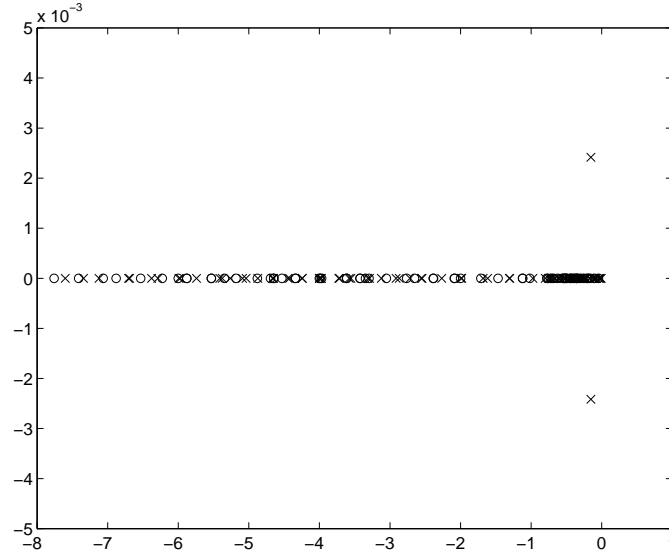


Figure 5.6: The eigenvalues of $\mathbf{A}_{21}\mathbf{A}_{12}$ for the problem shown in Fig. 5.5. The eigenvalues, for vacuum are indicated with circles, for the problem with the isotropic material, $\kappa_r = 16$, with crosses.

coarse grid boundary, but this does not pose a problem. Each eigenvalue of $\mathbf{A}_{21}\mathbf{A}_{12}$ is no longer negative real. One complex conjugate pair is present:

$$\lambda^2 = (-0.153 \pm 0.00242i) \left(\frac{c_0}{\Delta_f} \right)^2 \quad (5.44)$$

and the rest of the eigenvalues are negative real, with:

$$-7.602 \left(\frac{c_0}{\Delta_f} \right)^2 \leq \lambda^2 \leq -0.00706 \left(\frac{c_0}{\Delta_f} \right)^2 \quad (5.45)$$

The eigenvalues of \mathbf{A} that have moved away from the imaginary axis are:

$$\lambda = (\pm 0.00309 \pm 0.391i) \frac{c_0}{\Delta_f} \quad (5.46)$$

The eigenvalues of $\mathbf{A}_{21}\mathbf{A}_{12}$ are shown in Fig. 5.6 and are indicated with crosses (\times). The complex conjugate pair can clearly be distinguished.

These two very simple problems indicate the risk associated to subgridding as far as stability is concerned. This risk is clearly related to the spatial discretization of the grid. As stated before, the solutions proposed in the literature, [7], [8], [9], to employ spatially reciprocal subgridding schemes

have difficulties maintaining this reciprocity especially in the corner regions of the fine grid. The use of each spatially reciprocal subgridding technique involves some loss of accuracy, especially in the corner regions. Since we are especially interested in small subgridded subdomains, where corners are never far away, it is vital to have an accurate model, even in these fine grid corner regions. This shows the dilemma, when subgridding is involved, between accuracy and stability.

For the numerical examples in the following chapter, we selected the non-reciprocal subgridding techniques since the accuracy of the results was our main objective.

2.2.2 Artificial losses

A possible means to control this source of instability is adding some artificial losses. These losses force the eigenvalues into the right half plane. Remember, the poles are the negative eigenvalues. The system matrix of a non-reciprocal lossy system looks as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{D}_\theta^{-1/2} \mathbf{D}_\sigma \mathbf{D}_\theta^{-1/2} & \frac{c_0}{\Delta} \mathbf{D}_\theta^{-1/2} \mathbf{K}_1 \mathbf{D}_\kappa^{-1/2} \\ \frac{c_0}{\Delta} \mathbf{D}_\kappa^{-1/2} \mathbf{K}_2^T \mathbf{D}_\theta^{-1/2} & \mathbf{D}_\kappa^{-1/2} \mathbf{D}_{\sigma^*} \mathbf{D}_\kappa^{-1/2} \end{bmatrix} \quad (5.47)$$

When the diagonal blocks are of the form

$$\mathbf{A}_{11} = \mathbf{D}_\theta^{-1/2} \mathbf{D}_\sigma \mathbf{D}_\theta^{-1/2} = c \mathbf{I} \quad (5.48a)$$

$$\mathbf{A}_{22} = \mathbf{D}_\kappa^{-1/2} \mathbf{D}_{\sigma^*} \mathbf{D}_\kappa^{-1/2} = c \mathbf{I} \quad (5.48b)$$

the eigenvalues will have moved with a distance c to the right.

Since, for the TM-case $\mathbf{D}_\sigma \neq 0$, and for the TE-case $\mathbf{D}_{\sigma^*} \neq 0$ are not physical, it might be more interesting to include only one equation (5.48), and maintaining the other diagonal block to zero. This can also be a tool to force the unstable poles out of the right half plane. For $\mathbf{A}_{22} = 0$ and $\mathbf{A}_{11} = c \mathbf{I}$, and for an eigenvalue λ with corresponding eigenvector \mathbf{x} , one can write:

$$\begin{cases} c \mathbf{I} \mathbf{x}_1 + \mathbf{A}_{12} \mathbf{x}_2 = \lambda \mathbf{x}_1 \\ \mathbf{A}_{21} \mathbf{x}_1 = \lambda \mathbf{x}_2 \end{cases} \quad (5.49)$$

This leads to the following relation between the eigenvalues of $\mathbf{A}_{21} \mathbf{A}_{12}$ and λ :

$$\mathbf{A}_{21} \mathbf{A}_{12} \mathbf{x}_2 = (\lambda - c) \lambda \mathbf{x}_2 \quad (5.50)$$

The eigenvalues of $\mathbf{A}_{21} \mathbf{A}_{12}$ are no longer equal to λ^2 , but relate to $\lambda(\lambda - c)$. The same result is obtained for $\mathbf{A}_{11} = 0$ and $\mathbf{A}_{22} = c \mathbf{I}$. Calculations show,

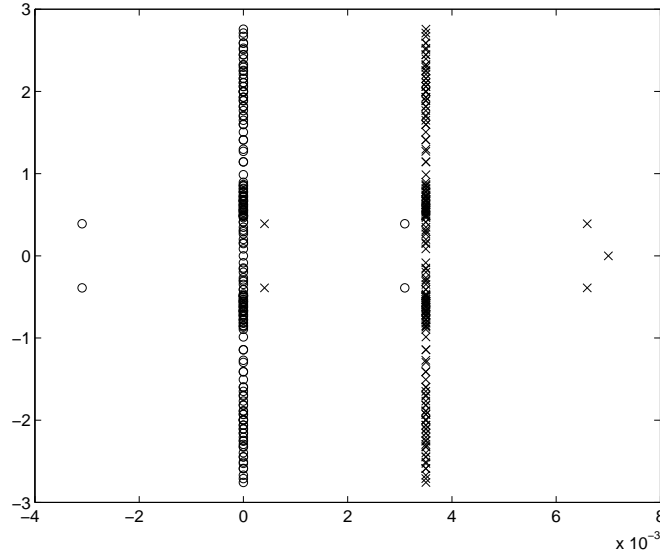


Figure 5.7: The eigenvalues of \mathbf{A} for the problem shown in Fig. 5.5, with the material $\kappa_r = 16$ present. The eigenvalues, without artificial losses, are indicated with circles, those with the artificial losses, $c = 0.007$, are indicated with crosses.

for an eigenvalue of $\mathbf{A}_{21}\mathbf{A}_{12}$ of the form $(a + bi)$, that

$$c \geq \sqrt{-\frac{b^2}{a}} \quad (5.51)$$

will result in stable eigenvalues for \mathbf{A} .

Consider again the simulation domain presented in Fig. 5.5, where the material, $\kappa_r = 16$ is present. The complex conjugate pair of eigenvalues can be forced away into the right half plane by choosing the losses, or the parameter c , high enough:

$$c \geq 0.00619 \frac{c_0}{\Delta_f} \quad (5.52)$$

The other eigenvalues, originally located on the imaginary axis, will also have moved by some c -related distance. In Fig. 5.7, the eigenvalues of \mathbf{A} are shown for two cases, the lossless case (symbol 'o') and the case with some artificial losses $c = 0.007$ (symbol 'x'). Clearly the eigenvalues have moved to the right. The instability is now under control since all eigenvalues are located in the right half plane.

For the TM-case the matrix $\mathbf{A}_{22} = c\mathbf{I}$ corresponds to a simulation domain where the losses in the vacuum material are

$$\sigma_e = \sqrt{\frac{\epsilon_0}{\mu_0}} \frac{0.007}{\Delta_f} = \frac{1.857 \cdot 10^{-5} \text{S}}{\Delta_f} \quad (5.53)$$

and the losses in the dielectric material are:

$$\sigma_e = 16 \sqrt{\frac{\epsilon_0}{\mu_0}} \frac{0.007}{\Delta_f} = \frac{2.971 \cdot 10^{-4} \text{S}}{\Delta_f} \quad (5.54)$$

where S signifies Siemens.

Often, when confronted with instability, some artificial losses will be used to resolve this instability. However for normal numerical simulations, the number of field variables is far too high to determine the artificial losses required, since it is prohibitively expensive to calculate the eigenvalues. Therefore a guess is made, hoping that the losses added will stabilize the simulation. Clearly, adding artificial losses comes with a cost: it corrupts the simulation in some way. It has to be hoped that the losses required do not influence the simulation too much. When the unstable poles are not too far into the right half plane it can be useful to force these poles, in this way, out of the right half plane.

2.2.3 Absorbing boundary conditions

Up to now, only problems were discussed where the problem space was truncated by a perfectly conducting material $P_z = 0$. This resulted in the very attractive block structure, e.g. for the lossy reciprocal case (5.15), and allowed to derive some interesting properties especially for the reciprocal case.

Unfortunately, the problem space is often not truncated by a perfectly conducting material, unless the problem space is so large, that the simulation is terminated before a wave, reflected at the boundary, has propagated back to the recording fields. This is hardly ever the case, and it is therefore essential to incorporate some kind of absorbing boundary condition [4]. An absorbing boundary condition terminates the grid in such a way that the lattice acts as if it was extended to infinity. Ideally, each wave impinging on such a boundary is not reflected and it simulates the extension of the problem domain to infinity.

Two categories of absorbing boundary conditions can be distinguished:

- Analytical absorbing boundary conditions, also called differential-equation based boundary conditions [13]. These absorbing boundary

conditions derive a discrete approximation of a one-way wave equation which allows waves only to propagate in the outside direction. This is then applied at the boundary of the FDTD grid. The most famous is the Mur absorbing boundary condition [14].

- Perfectly matched layer absorbing boundary conditions (PML), or material absorbing boundary conditions. The perfectly matched layer is an artificial material, attached at the outer lattice boundary. This material has two key properties: it does not reflect waves impinging on it, hence a perfectly matched material, and it has losses, causing waves propagating inside this material to decay. The material is then terminated again by a perfectly conducting material. The waves are absorbed inside this material. The perfectly matched layer was first presented in [15].

In the rest of this section a closer look will be taken at the consequences, as far as stability of the system is concerned, of using the first order Mur absorbing boundary condition [14] to terminate a grid. We will not go into aspects concerning accuracy and validity of the boundary condition, for that we refer to [4]. We only note that the Mur boundary condition is one of the first absorbing boundary conditions introduced and is conceptually easy, making it interesting to study. The Mur condition is a differential-equation based boundary condition.

The Mur absorbing boundary condition is based on the one-way wave equation in a direction orthogonal to the grid boundary. For the left side boundary, when the boundary medium is vacuum, this is for instance:

$$\frac{\partial O_y}{\partial x} = \frac{1}{c_0} \frac{\partial O_y}{\partial t} \quad \text{for } x = x_{\min} \quad (5.55)$$

Suppose that the left boundary of the grid is located at $x = 0$, then the spatially discretized version of this equation, obtained about the point $(\frac{1}{2}\Delta, (j + \frac{1}{2})\Delta)$ is:

$$\frac{O_y|_{1,j+\frac{1}{2}} - O_y|_{0,j+\frac{1}{2}}}{\Delta} = \frac{1}{2c_0} \left(\frac{d O_y|_{0,j+\frac{1}{2}}}{dt} + \frac{d O_y|_{1,j+\frac{1}{2}}}{dt} \right) \quad (5.56)$$

For the spatial derivative, a second order accurate central difference approximation was used, and the time derivative was approximated by a second order accurate average. In this way the field variables correspond to the normal locations inside the grid, for O_y this is $(i\Delta, (j + \frac{1}{2})\Delta)$.

A similar kind of equation can be formed for each tangential field variable located at the other boundaries. The nature of this equation is very

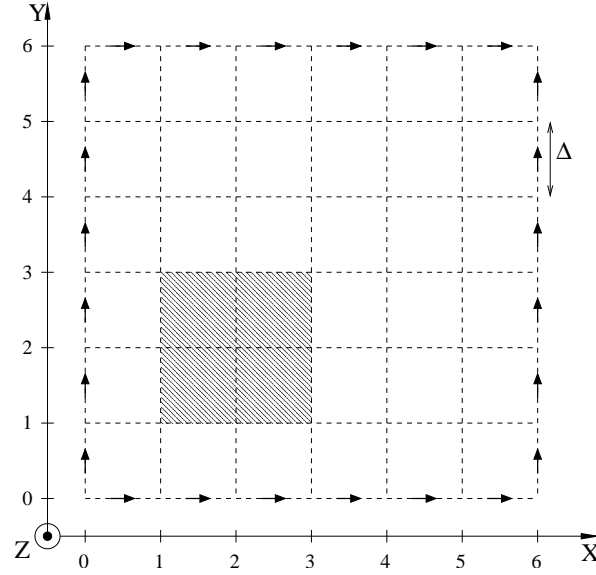


Figure 5.8: A small simulation problem, covering an area of 6 by 6 cells. The simulation domain is terminated by first order Mur absorbing boundary condition.

different from the normal FDTD equations. The equation (5.56), only involves O_y fields. In (5.56), it is possible to replace the term $dO_y|_{1,j+1/2}/dt$ by the corresponding FDTD equation. In this way, the system described by $\mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x}$, has a diagonal \mathbf{C} -matrix. However, the leading block \mathbf{G}_{11} of the matrix \mathbf{G} is no longer diagonal, since the equation (5.56) still has a term $O_y|_{1,j+1/2}$ disturbing the reciprocal form.

By means of a small example, the stability related consequences of these Mur absorbing boundary conditions are investigated, Fig. 5.8. The problem domain consists of a 6 by 6 grid terminated by a Mur absorbing boundary condition. The boundary field variables, where an equation similar to (5.56) is used, are indicated. The other field variables are not shown. The problem involves 84 O -type field variables and 36 P -type field variables.

The eigenvalues of $\mathbf{A} = \mathbf{C}^{-1}\mathbf{G}$ were investigated. As a first step the medium in the simulation domain was vacuum. In Fig. 5.9 the eigenvalues of \mathbf{A} have been plotted, 'o'. All eigenvalues have a positive real part, implying stability. It can be observed that the positive real part of most of these eigenvalues is quite large. This expresses the damping capacity of the system. Energy present in the system, can leave the system, waves are no longer confined.

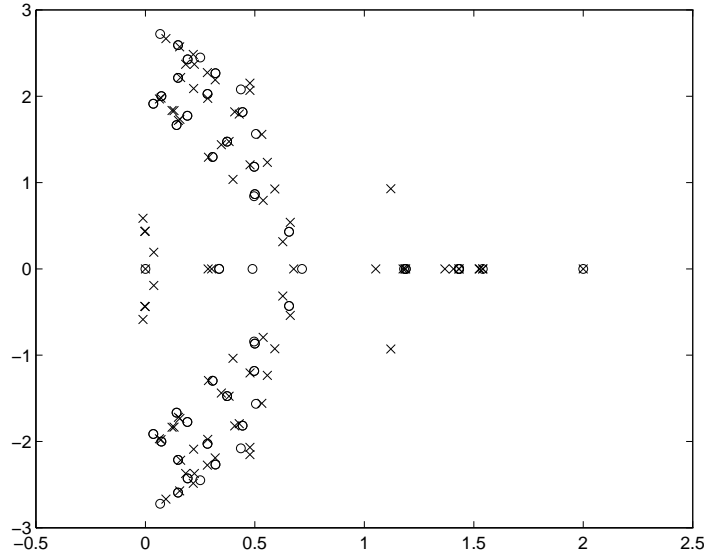


Figure 5.9: The eigenvalues for the simulation domain shown in Fig. 5.8, with vacuum are indicated with circles, those for the system with the isotropic material, $\kappa_r = 16$, are indicated with crosses.

By introducing an isotropic material, with $\kappa_r = 16$, inside the grid, the shaded region in Fig. 5.8, the eigenvalues are changed. In Fig. 5.9, the new eigenvalues are indicated by 'x'. Some eigenvalues now have a small negative real part and will cause instability to occur. In Fig. 5.10, a close up of the region around $\text{Re}[\lambda] = 0$ is shown. It shows that

$$\min(\text{Re}[\lambda]) = -0.011 \quad (5.57)$$

The simulation problem shown in Fig. 5.8 suffers, due to the absorbing boundary condition from instability. The grid in Fig. 5.8, has unstable poles.

Although the FDTD method would not be of much interest without absorbing boundary conditions, it has to be noted that these absorbing boundary conditions are a possible source of instability. This illustrates the conflict between special FDTD methods, enhancing the versatility of the method, and basic requirements, related to stability. The example illustrates that even though stability cannot be guaranteed in most simulations, this does not mean that these simulations can not lead to meaningful results. For several years the Mur condition was the only absorbing boundary condition and experience in the field has shown that it is not very

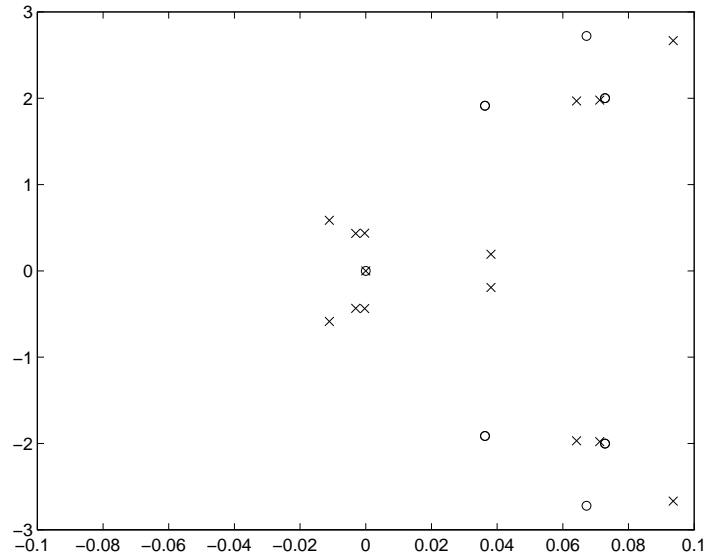


Figure 5.10: A close up of Fig. 5.9. It shows more clearly that some eigenvalues have a negative real part or correspondingly have unstable poles.

susceptible to instability. This puts the problems related to subgridding into perspective.

3 ROM RELATED STABILITY

3.1 The reduced order model

Where the spatial discretization is a first step for each FDTD method, a second step, in the methods presented in this work, is formed by the reduced order modeling (ROM) technique applied to a subsystem describing a subdomain. In the previous section the stability related spatial discretization was investigated. In this section, a closer look will be taken at what happens when a part of a system is replaced by a reduced model. In Chapter 4 a number of reduced order modeling techniques were presented. It was mentioned that especially the passivity preserving reduced order modeling techniques are of interest.

Let us start by considering a system, representing a subdomain, of the

following form:

$$\left\{ \begin{array}{l} \begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_\sigma & \frac{c_0}{\Delta} \mathbf{K} \\ -\frac{c_0}{\Delta} \mathbf{K}^T & \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u} \\ \mathbf{y} = \begin{bmatrix} \mathbf{L}_1^T & \mathbf{L}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \end{array} \right. \quad (5.58)$$

The subdomain then only has a regular grid and no absorbing boundary conditions or subgridding inside the subdomain. The subdomain is connected to a neighbouring or surrounding grid through the input variables \mathbf{u} and the output variables \mathbf{y} .

In this form, matrix \mathbf{C} is diagonal with positive definite blocks and matrix \mathbf{G} has two positive semidefinite diagonal blocks and the other blocks are skew symmetric. This is very similar to the passive form of circuits (4.64). It is therefore logical to use the projection based ROM techniques that preserve passivity for the RLC circuits. The matrices of the reduced system are related to the matrices of the original system through:

$$\tilde{\mathbf{C}} = \mathbf{U}^T \mathbf{C} \mathbf{U} \quad (5.59a)$$

$$\tilde{\mathbf{G}} = \mathbf{U}^T \mathbf{G} \mathbf{U} \quad (5.59b)$$

$$\tilde{\mathbf{B}} = \mathbf{U}^T \mathbf{B} \quad (5.59c)$$

$$\tilde{\mathbf{L}} = \mathbf{U}^T \mathbf{L} \quad (5.59d)$$

As a first step the reduced system, and the poles of this system, will be considered $\tilde{\mathbf{C}}\dot{\mathbf{z}} = -\tilde{\mathbf{G}}\mathbf{z}$.

Since the matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{D}_\theta & 0 \\ 0 & \mathbf{D}_\kappa \end{bmatrix} \quad (5.60)$$

is positive definite, the following condition holds

$$\mathbf{x}^T \mathbf{C} \mathbf{x} > 0 \quad \forall \mathbf{x} \neq 0 \quad (5.61)$$

The reduced matrix $\tilde{\mathbf{C}}$ is also positive definite since

$$\mathbf{y}^T \tilde{\mathbf{C}} \mathbf{y} = \mathbf{y}^T \mathbf{U}^T \mathbf{C} \mathbf{U} \mathbf{y} \quad (5.62)$$

$$= (\mathbf{U} \mathbf{y})^T \mathbf{C} (\mathbf{U} \mathbf{y}) \quad (5.63)$$

is always positive (5.61). Note that the vector $\mathbf{U} \mathbf{y}$ is always different from zero since \mathbf{U} is a basis. A symmetric positive definite matrix, as $\tilde{\mathbf{C}}$, can, using the Cholesky factorization [12], be written as $\tilde{\mathbf{M}} \tilde{\mathbf{M}}^T$. The poles of the reduced system then correspond to the negative eigenvalues of $\tilde{\mathbf{A}} = \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{G}} \tilde{\mathbf{M}}^{-T}$. This matrix can easily be written as the sum of a symmetric

part $\tilde{\mathbf{S}}$ and a skew symmetric part $\tilde{\mathbf{R}}$, by considering the symmetric and skew symmetric part of \mathbf{G} :

$$\tilde{\mathbf{S}} = \tilde{\mathbf{M}}^{-1} \mathbf{U} \left(\frac{\mathbf{G} + \mathbf{G}^T}{2} \right) \mathbf{U} \tilde{\mathbf{M}}^{-T} \quad (5.64a)$$

$$\tilde{\mathbf{R}} = \tilde{\mathbf{M}}^{-1} \mathbf{U} \left(\frac{\mathbf{G} - \mathbf{G}^T}{2} \right) \mathbf{U} \tilde{\mathbf{M}}^{-T} \quad (5.64b)$$

We are only interested in the real parts of the eigenvalues of the matrix $\tilde{\mathbf{A}}$, since this determines the stability of the system. As explained before the symmetric part $\tilde{\mathbf{S}}$ of the matrix $\tilde{\mathbf{A}}$ determines the bounds for these real parts. Since $1/2(\mathbf{G} + \mathbf{G}^T) \geq 0$ only contains the diagonal blocks of \mathbf{G} it is positive semidefinite and consequently the matrix $\tilde{\mathbf{S}} \geq 0$ is also positive semidefinite and has no negative real eigenvalues. In this way it can be concluded that the reduced system generated using (5.59) does not have poles in the right half plane.

In a similar way it can be shown that a system without unstable poles can be generated starting from:

$$\begin{cases} \dot{\mathbf{x}}' = -\mathbf{G}' \mathbf{x}' + \mathbf{B}' \mathbf{u} \\ \mathbf{y} = \mathbf{L}'^T \mathbf{x}' \end{cases} \quad (5.65)$$

where

$$\mathbf{C}' = \mathbf{I} \quad (5.66a)$$

$$\mathbf{G}' = \mathbf{C}^{-1/2} \mathbf{G} \mathbf{C}^{-1/2} \quad (5.66b)$$

$$\mathbf{B}' = \mathbf{C}^{-1/2} \mathbf{B} \quad (5.66c)$$

$$\mathbf{L}' = \mathbf{C}^{-1/2} \mathbf{L} \quad (5.66d)$$

The advantage in that case is that the matrix \mathbf{C}' and $\tilde{\mathbf{C}}$ remain identity matrices.

A third possible way could be to start from

$$\mathbf{C}' = \mathbf{I} \quad (5.67a)$$

$$\mathbf{G}' = \mathbf{C}^{-1} \mathbf{G} \quad (5.67b)$$

$$\mathbf{B}' = \mathbf{C}^{-1} \mathbf{B} \quad (5.67c)$$

$$\mathbf{L}' = \mathbf{L} \quad (5.67d)$$

However the specific structure of $\mathbf{C}^{-1} \mathbf{G}$ no longer corresponds to the structure of \mathbf{G} . Numerical examples show that this gives rise to unstable poles after reduction and is therefore not recommended.

3.2 Spatially reciprocal grids

Poles of an entire system are not only determined by the poles of a subsystem, meaning a stable subsystem not necessarily involves a stable system. However, vice versa when a subsystem has unstable poles, it is unlikely that, when this subsystem is used in a larger system, the larger system will be stable. In this section we have a look at a system, derived from a spatially reciprocal grid, where a part has been replaced by a projection based reduced order model.

Consider a system divided into two subsystems:

$$\begin{cases} \mathbf{C}_1 \dot{\mathbf{x}}_1 = -\mathbf{G}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{u}_1 \\ \mathbf{y}_1 = \mathbf{L}_1^T \mathbf{x}_1 \end{cases} \quad (5.68a)$$

$$\begin{cases} \mathbf{C}_2 \dot{\mathbf{x}}_2 = -\mathbf{G}_2 \mathbf{x}_2 + \mathbf{B}_2 \mathbf{u}_2 \\ \mathbf{y}_2 = \mathbf{L}_2^T \mathbf{x}_2 \end{cases} \quad (5.68b)$$

where

$$\mathbf{C}_i = \begin{bmatrix} \mathbf{D}_{\theta,i} & 0 \\ 0 & \mathbf{D}_{\kappa,i} \end{bmatrix} \quad (5.69a)$$

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{D}_{\sigma,i} & \frac{c_0}{\Delta} \mathbf{K}_i \\ -\frac{c_0}{\Delta} \mathbf{K}_i^T & \mathbf{D}_{\sigma^*,i} \end{bmatrix} \quad (5.69b)$$

and since spatial reciprocity is assumed, the relation $\mathbf{B}_1 \mathbf{L}_2^T = -(\mathbf{B}_2 \mathbf{L}_1)^T$ can be added.

Replacing one subdomain, say subdomain 1, by a reduced order model:

$$\begin{cases} \tilde{\mathbf{C}}_1 \dot{\mathbf{z}} = -\tilde{\mathbf{G}}_1 \mathbf{z} + \tilde{\mathbf{B}}_1 \mathbf{u}_1 \\ \mathbf{y}_1 = \tilde{\mathbf{L}}_1^T \mathbf{z} \end{cases} \quad (5.70)$$

where the new matrices are related to the old matrices by (5.59), and by reminding again that $\mathbf{u}_1 = \mathbf{y}_2$ and $\mathbf{u}_2 = \mathbf{y}_1$, it is possible to write the entire system as:

$$\begin{bmatrix} \tilde{\mathbf{C}}_1 & 0 \\ 0 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{z}} \\ \dot{\mathbf{x}}_2 \end{bmatrix} = - \begin{bmatrix} \tilde{\mathbf{G}}_1 & \tilde{\mathbf{B}}_1 \mathbf{L}_2^T \\ \mathbf{B}_2 \tilde{\mathbf{L}}_1^T & \mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{x}_2 \end{bmatrix} \quad (5.71)$$

By introducing the Cholesky decomposition of $\tilde{\mathbf{C}}_1 = \tilde{\mathbf{M}}_1 \tilde{\mathbf{M}}_1^T$, the real parts of the eigenvalues of

$$\begin{aligned} \begin{bmatrix} \tilde{\mathbf{M}}_1^{-1} & 0 \\ 0 & \mathbf{C}_2^{-1/2} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{G}}_1 & \tilde{\mathbf{B}}_1 \mathbf{L}_2^T \\ \mathbf{B}_2 \tilde{\mathbf{L}}_1^T & \mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{M}}_1^{-T} & 0 \\ 0 & \mathbf{C}_2^{-1/2} \end{bmatrix} \\ = \begin{bmatrix} \tilde{\mathbf{M}}_1^{-1} \tilde{\mathbf{G}}_1 \tilde{\mathbf{M}}_1^{-T} & \tilde{\mathbf{M}}_1^{-1} \tilde{\mathbf{B}}_1 \mathbf{L}_2^T \mathbf{C}_2^{-1/2} \\ \mathbf{C}_2^{-1/2} \mathbf{B}_2 \tilde{\mathbf{L}}_1 \tilde{\mathbf{M}}_1^{-T} & \mathbf{C}_2^{-1/2} \mathbf{G}_2 \mathbf{C}_2^{-1/2} \end{bmatrix} \end{aligned} \quad (5.72)$$

determine the stability of the system.

The two off-diagonal blocks are still each others counterpart:

$$\left(\tilde{\mathbf{M}}_1^{-1} \tilde{\mathbf{B}}_1 \mathbf{L}_2^T \mathbf{C}_2^{-1/2}\right)^T = \left(\tilde{\mathbf{M}}_1^{-1} \mathbf{U}^T \mathbf{B}_1 \mathbf{L}_2^T \mathbf{C}_2^{-1/2}\right)^T \quad (5.73)$$

$$= \mathbf{C}_2^{-1/2} \left(\mathbf{B}_1 \mathbf{L}_2^T\right)^T \mathbf{U} \tilde{\mathbf{M}}_1^{-T} \quad (5.74)$$

$$= -\mathbf{C}_2^{-1/2} \mathbf{B}_2 \mathbf{L}_1^T \mathbf{U} \tilde{\mathbf{M}}_1^{-T} \quad (5.75)$$

$$= -\mathbf{C}_2^{-1/2} \mathbf{B}_2 \tilde{\mathbf{L}}_1^T \tilde{\mathbf{M}}_1^{-T} \quad (5.76)$$

which means that the symmetric part of the system matrix (5.72) can be written as:

$$\frac{1}{2} \begin{bmatrix} \tilde{\mathbf{M}}_1^{-1} \mathbf{U}^T (\mathbf{G}_1 + \mathbf{G}_1^T) \mathbf{U} \tilde{\mathbf{M}}_1^{-T} & 0 \\ 0 & \mathbf{C}_2^{-1/2} (\mathbf{G}_2 + \mathbf{G}_2^T) \mathbf{C}_2^{-1/2} \end{bmatrix} \quad (5.77)$$

Both diagonal blocks are positive semidefinite, meaning that the entire matrix is positive semidefinite. Hence, the system matrix has no eigenvalues in the left half plane. This means that for a reciprocal system, any part can be reduced and it will not affect the stability of the overall system. This result confirms the stability preserving capabilities of the projection based ROM techniques.

3.3 Spatially nonreciprocal grids

It is our intention to apply a ROM technique to a spatially nonreciprocal subdomain. This subdomain, however will often be part of a nonreciprocal grid. Considering again the two systems of the previous section (5.68), where still $\mathbf{u}_1 = \mathbf{y}_2$ and $\mathbf{u}_2 = \mathbf{y}_1$, but now $\mathbf{B}_1 \mathbf{L}_2^T \neq -(\mathbf{B}_2 \mathbf{L}_1^T)^T$. Two subdomains have been identified, where both subdomains, when inspected separately, are reciprocal. This is, when applied to subgridding, the fine grid and the coarse grid. The connection between both subdomains, or mathematically \mathbf{B}_1 , \mathbf{L}_1 , \mathbf{B}_2 and \mathbf{L}_2 , causes the loss of reciprocity. When one subsystem is replaced by a reduced order model, say subsystem 1, it is now no longer possible to determine whether the system is stable or not. This is mainly due to the off diagonal blocks of the system matrix (5.72):

$$\left(\tilde{\mathbf{M}}_1^{-1} \tilde{\mathbf{B}}_1 \mathbf{L}_2^T \mathbf{C}_2^{-1/2}\right)^T \neq -\mathbf{C}_2^{-1/2} \mathbf{B}_2 \tilde{\mathbf{L}}_1^T \tilde{\mathbf{M}}_1^{-T} \quad (5.78)$$

4 TIME DISCRETIZATION RELATED STABILITY

In this last section, a closer look is taken at the consequences of time discretization. This will be done for the different time discretization schemes discussed in Chapter 3: the standard FDTD method, the ADI-FDTD method and the new subdomain method. The subdomain FDTD method is a combination of two extremes: an explicit method as the normal FDTD equations on the one hand, and an implicit method, as is for example the ADI-FDTD method, on the other hand. The subdomain method has a lot in common with the FDTD method, especially the leapfrog time stepping. With the ADI-FDTD method, the link is not so strong, since the implicit part in the subdomain FDTD method is obtained by calculating a matrix inversion. The consequences of explicit and implicit methods on the stability and more specifically on the maximum time step will be discussed here.

4.1 Standard FDTD method

4.1.1 General aspects

The most famous stability criterion derived up to now, which also puts a limit on the maximum time step that can be used in an FDTD simulation is the so-called Courant limit [1]. For the 2D case, with $\Delta_x = \Delta_y = \Delta$, the Courant condition is:

$$\Delta_t \leq \frac{\Delta}{\sqrt{2}c_0} \quad (5.79)$$

This relation is valid for problems extending to infinity, no boundary conditions are involved, and for a homogeneous medium. The Courant condition has proven to be very important, since it puts an upper bound on the time step that can be chosen for a specific simulation. Von Neumann analysis was used to obtain this stability condition. Furthermore, the Von Neumann analysis, leads to a numerical dispersion relation giving an expression for the phase velocity of the numerical waves propagating in the grid, as a function of angle of propagation and space step [4]. We will derive an expression for the maximum time step in the FDTD method, similar to (5.79), in a grid truncated by perfectly conducting material. However, this will be done based on the system matrices and the eigenvalues of it. This does not lead to the dispersion relation, but sheds a new light on the stability issue in the FDTD method.

Let us start with the FDTD update equations, for the lossy case, in matrix form (3.21):

$$\begin{bmatrix} \mathbf{D}_\theta + \frac{\Delta t}{2} \mathbf{D}_\sigma & 0 \\ -k\mathbf{K}^T & \mathbf{D}_\kappa + \frac{\Delta t}{2} \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n+1/2} \\ \mathbf{p}|^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_\theta - \frac{\Delta t}{2} \mathbf{D}_\sigma & -k\mathbf{K} \\ 0 & \mathbf{D}_\kappa - \frac{\Delta t}{2} \mathbf{D}_{\sigma^*} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n-1/2} \\ \mathbf{p}|^n \end{bmatrix} \quad (5.80)$$

where

$$k = \frac{c_0 \Delta t}{\Delta} \quad (5.81)$$

These are the FDTD equations for a grid truncated by a perfectly conducting material.

First of all note that (5.80) can be written as

$$(\mathbf{E} + \mathbf{F})\mathbf{x}|^{n+1} = (\mathbf{E} - \mathbf{F})\mathbf{x}|^n \quad (5.82)$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{D}_\theta & -\frac{k}{2}\mathbf{K} \\ -\frac{k}{2}\mathbf{K}^T & \mathbf{D}_\kappa \end{bmatrix} \quad (5.83a)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\Delta t}{2} \mathbf{D}_\sigma & \frac{k}{2} \mathbf{K} \\ -\frac{k}{2} \mathbf{K}^T & \frac{\Delta t}{2} \mathbf{D}_{\sigma^*} \end{bmatrix} \quad (5.83b)$$

$$\mathbf{x}|^n = \begin{bmatrix} \mathbf{o}|^{n-1/2} \\ \mathbf{p}|^n \end{bmatrix} \quad (5.83c)$$

In this way, the iteration matrix is:

$$(\mathbf{E} + \mathbf{F})^{-1} (\mathbf{E} - \mathbf{F}) \quad (5.84)$$

and this matrix expresses how the new values $\mathbf{x}|^{n+1}$ can be calculated from the old values $\mathbf{x}|^n$. The eigenvalues of the iteration matrix (5.84) determine the stability of the algorithm. Determining when an eigenvalue μ of (5.84), has an absolute value larger than one, $|\mu| > 1$, determines the stability of the time stepping algorithm. Since eigenvalues are invariant under similarity transformations, these, when \mathbf{E} is nonsingular, are also the eigenvalues of

$$(\mathbf{I} + \mathbf{E}^{-1}\mathbf{F})^{-1} (\mathbf{I} - \mathbf{E}^{-1}\mathbf{F}) \quad (5.85)$$

There is an easy relation between the eigenvalues, λ , of $\mathbf{E}^{-1}\mathbf{F}$ and μ , an eigenvalue of (5.85):

$$\mu = \frac{1 + \lambda}{1 - \lambda} \quad (5.86)$$

This is a conformal mapping of the right half plane, considering λ , onto the unit disk, considering μ . The problem of finding $|\mu| \leq 1$ is similar to finding $\text{Re}[\lambda] \geq 0$. This bilinear transformation has already been encountered in Chapter 4, in the discussion of the Laguerre-SVD reduced order modeling technique. The variation of variable introduced at that point was (4.70):

$$\sigma = \frac{s - \alpha}{s + \alpha} \quad (5.87)$$

For λ an eigenvalue of $\mathbf{E}^{-1}\mathbf{F}$ and \mathbf{z} the associated eigenvector, then:

$$\mathbf{E}^{-1}\mathbf{F}\mathbf{z} = \lambda\mathbf{z} \quad (5.88)$$

or

$$\mathbf{F}\mathbf{z} = \lambda\mathbf{E}\mathbf{z} \quad (5.89)$$

By left multiplying (5.89) by \mathbf{z}^H it is clear that:

$$\mathbf{z}^H\mathbf{F}\mathbf{z} = \lambda\mathbf{z}^H\mathbf{E}\mathbf{z} \quad (5.90)$$

Taking the self-adjoint of this last equation then gives:

$$\mathbf{z}^H\mathbf{F}^T\mathbf{z} = \lambda^*\mathbf{z}^H\mathbf{E}^T\mathbf{z} \quad (5.91)$$

Since \mathbf{E} and \mathbf{F} are real matrices, $\mathbf{E}^H = \mathbf{E}^T$ and $\mathbf{F}^H = \mathbf{F}^T$. Furthermore, in our case the matrix \mathbf{E} is symmetric, $\mathbf{E}^T = \mathbf{E}$. By adding the last two equations and dividing by 2, the following is obtained:

$$\frac{1}{2}\mathbf{z}^H(\mathbf{F}^T + \mathbf{F})\mathbf{z} = \text{Re}(\lambda)\mathbf{z}^H\mathbf{E}\mathbf{z} \quad (5.92)$$

Having a closer look at the left hand side of (5.92), it can easily be seen that

$$\frac{1}{2}(\mathbf{F}^T + \mathbf{F}) = \begin{bmatrix} \frac{\Delta t}{2}\mathbf{D}_\sigma & 0 \\ 0 & \frac{\Delta t}{2}\mathbf{D}_{\sigma^*} \end{bmatrix} \quad (5.93)$$

is positive semidefinite. Hence the left hand side of (5.92) is always zero or positive. With this result, it suffices to determine when the symmetric matrix $\mathbf{E} > 0$ is positive definite, because then $\mathbf{z}^H\mathbf{E}\mathbf{z} > 0$ for all \mathbf{z} and as a consequence $\text{Re}(\lambda) \geq 0$, which proves the stability. The matrix is not allowed to be positive semidefinite, $\mathbf{E} \geq 0$, since \mathbf{E} needs to be nonsingular.

The matrix \mathbf{E} is the only matrix that determines the stability of the FDTD time stepping algorithm. Equation (5.83a), makes clear that \mathbf{E} is independent of the losses present inside the problem space. Therefore losses do not play a role in the stability of the explicit FDTD time stepping algorithm. This is in agreement with [17], where, based on the Von Neumann analysis, the stability criterion for lossy dielectrics was determined.

4.1.2 The vacuum case: boundary tangential P fields zero

When dealing with a problem where only vacuum is involved, $\theta_r = 1$ and $\kappa_r = 1$, the 'large' singular value decomposition of \mathbf{K} , as in Section 2.2.1, can help:

$$\mathbf{K} = \mathbf{U}^T \Sigma \mathbf{V} \quad (5.94)$$

where $\mathbf{K}, \Sigma \in \mathbb{R}^{n_1 \times n_2}$, $\mathbf{U} \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times n_2}$. The matrix Σ is diagonal

$$[\Sigma]_{ii} = \sigma_i \quad \text{for } i = 1, 2, \dots, n_2 \quad (5.95)$$

and contains the singular values of \mathbf{K} . It is possible, since $\mathbf{D}_\theta = \mathbf{I}$ and $\mathbf{D}_\kappa = \mathbf{I}$, to write \mathbf{E} as:

$$\mathbf{E} = \begin{bmatrix} \mathbf{U}^T & 0 \\ 0 & \mathbf{V}^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\frac{k}{2}\Sigma \\ -\frac{k}{2}\Sigma & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{V} \end{bmatrix} \quad (5.96)$$

Employing the same permutation matrix \mathbf{P} as in Section 2.2.1, this becomes:

$$\mathbf{E} = \begin{bmatrix} \mathbf{U}^T & 0 \\ 0 & \mathbf{V}^T \end{bmatrix} \mathbf{P} \Omega \mathbf{P}^T \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{V} \end{bmatrix} \quad (5.97)$$

where Ω is a block diagonal matrix, with on the diagonal n_2 blocks of size 2×2 , and $(n_1 - n_2)$ ones:

$$\Omega = \text{diag} \left(\begin{bmatrix} 1 & -\frac{k}{2}\sigma_1 \\ -\frac{k}{2}\sigma_1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -\frac{k}{2}\sigma_2 \\ -\frac{k}{2}\sigma_2 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 1 & -\frac{k}{2}\sigma_{n_2} \\ -\frac{k}{2}\sigma_{n_2} & 1 \end{bmatrix}, 1, 1, \dots, 1 \right) \quad (5.98)$$

Since \mathbf{E} is similar to Ω , the eigenvalues are identical. The eigenvalues for one of these blocks is:

$$\lambda \left(\begin{bmatrix} 1 & -\frac{k\sigma_i}{2} \\ -\frac{k\sigma_i}{2} & 1 \end{bmatrix} \right) = 1 \pm \frac{k\sigma_i}{2} \quad (5.99)$$

Suppose the singular values have been ordered, with σ_1 the largest singular value belonging to \mathbf{K} , the condition for stability is then:

$$k = \frac{c_0 \Delta t}{\Delta} < \frac{2}{\sigma_1} \quad (5.100)$$

This relation was also derived in [18].

The largest singular value σ_1 , is only dependent on \mathbf{K} . Therefore the relation (5.100), can always be made true for a specific space step, the factor Δ , and a specific grid, the factor σ_1 , by reducing the time step. Choosing Δt small enough results in a stable FDTD algorithm. Furthermore, for certain

grids, it is possible to find an expression for σ_1 . Note that the singular values of \mathbf{K} are related to the eigenvalues of $\mathbf{K}^T \mathbf{K}$:

$$\mathbf{K}^T \mathbf{K} = \mathbf{V}^T \Sigma^T \mathbf{U} \mathbf{U}^T \Sigma \mathbf{V} \quad (5.101)$$

$$= \mathbf{V}^T \Sigma^T \Sigma \mathbf{V} \quad (5.102)$$

The eigenvalues of $\mathbf{K}^T \mathbf{K}$ are σ_i^2 , for $i = 1, 2, \dots, n_2$.

Consider the explicit form of \mathbf{K} derived in Section 4.3.1.4 of Chapter 2. There it was shown that for a grid consisting of $n_x \times n_y$ Yee cells terminated by $P_z = 0$ outside of these cells, the matrix \mathbf{K} could be written as (2.68):

$$\mathbf{K} = \begin{bmatrix} \mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y} \\ -\mathbf{W}_{n_x} \otimes \mathbf{I}_{n_y} \end{bmatrix} \quad (5.103)$$

where $\mathbf{W}_s \in \mathbb{R}^{(s+1) \times s}$ and

$$[\mathbf{W}_s]_{ij} = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } (i+1) = j \\ 0 & \text{else} \end{cases} \quad (5.104)$$

The matrix $\mathbf{K}^T \mathbf{K}$ can now be explicitly written as:

$$\mathbf{K}^T \mathbf{K} = (\mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y}^T) (\mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y}) + (\mathbf{W}_{n_x}^T \otimes \mathbf{I}_{n_y}) (\mathbf{W}_{n_x} \otimes \mathbf{I}_{n_y}) \quad (5.105)$$

$$= \mathbf{I}_{n_x} \otimes (\mathbf{W}_{n_y}^T \mathbf{W}_{n_y}) + (\mathbf{W}_{n_x}^T \mathbf{W}_{n_x}) \otimes \mathbf{I}_{n_y} \quad (5.106)$$

In this expression the following properties of the Kronecker product were used:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD} \quad (5.107)$$

$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T \quad (5.108)$$

Before continuing, we introduce the following special class of tridiagonal matrices $\mathbf{T}_s(a) \in \mathbb{R}^{s \times s}$:

$$[\mathbf{T}_s(a)]_{ij} = \begin{cases} a & \text{for } i = j \\ -1 & \text{for } |i - j| = 1 \\ 0 & \text{else} \end{cases} \quad (5.109)$$

The subindex of $\mathbf{T}_s(a)$ refers to the dimension of the matrix. Each diagonal element of $\mathbf{T}_s(a)$ is a , and the element above and below this diagonal is -1 . In [19], the eigenvalues of this frequently appearing tridiagonal matrix, $\mathbf{T}_s(a)$, are calculated:

$$a + 2 \cos\left(\frac{i\pi}{s+1}\right) \quad \text{for } i = 1, 2, \dots, s \quad (5.110)$$

It can easily be verified that:

$$\mathbf{W}_{n_x}^T \mathbf{W}_{n_x} = \mathbf{T}_{n_x}(2) \quad (5.111a)$$

$$\mathbf{W}_{n_y}^T \mathbf{W}_{n_y} = \mathbf{T}_{n_y}(2) \quad (5.111b)$$

The matrix (5.106), is then:

$$\begin{bmatrix} \mathbf{T}_{n_y}(4) & -\mathbf{I} & & & & \\ & -\mathbf{I} & \mathbf{T}_{n_y}(4) & -\mathbf{I} & & \\ & & -\mathbf{I} & \mathbf{T}_{n_y}(4) & -\mathbf{I} & \\ & & & \ddots & \ddots & \ddots \\ & & & & -\mathbf{I} & \mathbf{T}_{n_y}(4) & -\mathbf{I} \\ & & & & & -\mathbf{I} & \mathbf{T}_{n_y}(4) \end{bmatrix} \quad (5.112)$$

By introducing the eigendecomposition of $\mathbf{T}_{n_y}(4) = \mathbf{Q}^{-1}\Lambda\mathbf{Q}$, where Λ is diagonal and contains all the eigenvalues of $\mathbf{T}_{n_y}(4)$, it is clear that this matrix is similar to:

$$\begin{bmatrix} \Lambda & -\mathbf{I} & & & & \\ & -\mathbf{I} & \Lambda & -\mathbf{I} & & \\ & & -\mathbf{I} & \Lambda & -\mathbf{I} & \\ & & & \ddots & \ddots & \ddots \\ & & & & -\mathbf{I} & \Lambda & -\mathbf{I} \\ & & & & & -\mathbf{I} & \Lambda \end{bmatrix} \quad (5.113)$$

By using the appropriate permutation matrix \mathbf{P} , each of the eigenvalues of Λ can be grouped in one diagonal block, resulting in:

$$\begin{bmatrix} \mathbf{T}_{n_x}(\lambda_1) & & & & \\ & \mathbf{T}_{n_x}(\lambda_2) & & & \\ & & \mathbf{T}_{n_x}(\lambda_3) & & \\ & & & \ddots & \\ & & & & \mathbf{T}_{n_x}(\lambda_{n_y}) \end{bmatrix} \quad (5.114)$$

where

$$\lambda_i = 4 + 2 \cos \left(\frac{i\pi}{n_y + 1} \right) \quad \text{for } i = 1, 2, \dots, n_y \quad (5.115)$$

In this way, the eigenvalues of $\mathbf{K}^T\mathbf{K}$ have been calculated:

$$4 + 2 \cos \left(\frac{i\pi}{n_y + 1} \right) + 2 \cos \left(\frac{j\pi}{n_x + 1} \right) \quad \text{for } \begin{cases} i = 1, 2, \dots, n_y \\ j = 1, 2, \dots, n_x \end{cases} \quad (5.116)$$

The largest eigenvalue, corresponding to the square of the largest singular value of \mathbf{K} is, for $i = j = 1$:

$$4 + 2 \cos\left(\frac{\pi}{n_y + 1}\right) + 2 \cos\left(\frac{\pi}{n_x + 1}\right) \quad (5.117)$$

The stability condition for the FDTD time stepping algorithm is then:

$$k = \frac{c_0 \Delta_t}{\Delta} < \frac{2}{\sqrt{4 + 2 \cos\left(\frac{\pi}{n_y + 1}\right) + 2 \cos\left(\frac{\pi}{n_x + 1}\right)}} \quad (5.118)$$

For very large n_x , $\cos\left(\frac{\pi}{n_x + 1}\right)$ goes to 1, similarly for n_y . The stability condition for an infinitely large grid then becomes:

$$k = \frac{c_0 \Delta_t}{\Delta} < \frac{2}{\sqrt{8}} = \frac{1}{\sqrt{2}} \quad (5.119)$$

This result is the Courant condition. The stability condition (5.118) for the FDTD method, derived in this way, is just less strict than the Courant stability condition. For a grid extending to infinity, both results agree. Recently, in [20], a similar result, although the calculation steps were omitted, was shown for $\Delta_x \neq \Delta_y$.

4.1.3 Isotropic materials

When dealing with vacuum, a condition for which \mathbf{E}

$$\mathbf{E} = \begin{bmatrix} \mathbf{I} & -\frac{k}{2}\mathbf{K} \\ -\frac{k}{2}\mathbf{K}^T & \mathbf{I} \end{bmatrix} \quad (5.120)$$

is positive definite was derived in the previous section. However when some isotropic material is present, the matrix \mathbf{E} changes to:

$$\mathbf{E} = \begin{bmatrix} \mathbf{D}_\theta & -\frac{k}{2}\mathbf{K} \\ -\frac{k}{2}\mathbf{K}^T & \mathbf{D}_\kappa \end{bmatrix} \quad (5.121)$$

where \mathbf{D}_θ and \mathbf{D}_κ are positive and have diagonal elements larger than or equal to one. The positive definiteness of (5.120) implies $\mathbf{x}^T \mathbf{E} \mathbf{x} > 0$ for every \mathbf{x} . Making use of the block structure of \mathbf{E} , this means, for $\mathbf{x}^T = [\mathbf{x}_1^T \quad \mathbf{x}_2^T]$,

$$\mathbf{x}_1^T \mathbf{x}_1 - \frac{k}{2} \mathbf{x}_1^T \mathbf{K} \mathbf{x}_2 - \frac{k}{2} \mathbf{x}_2^T \mathbf{K}^T \mathbf{x}_1 + \mathbf{x}_2^T \mathbf{x}_2 > 0 \quad (5.122)$$

Making use of $[\mathbf{D}_\theta]_{ii} \geq 1$ and $[\mathbf{D}_\kappa]_{ii} \geq 1$, the following inequalities hold:

$$\mathbf{x}_1^T \mathbf{D}_\theta \mathbf{x}_1 \geq \mathbf{x}_1^T \mathbf{x}_1 \quad (5.123a)$$

$$\mathbf{x}_2^T \mathbf{D}_\kappa \mathbf{x}_2 \geq \mathbf{x}_2^T \mathbf{x}_2 \quad (5.123b)$$

Therefore, for every $\mathbf{x}^T = [\mathbf{x}_1^T \quad \mathbf{x}_2^T]$ the following inequality holds

$$\mathbf{x}_1^T \mathbf{D}_\theta \mathbf{x}_1 - \frac{k}{2} \mathbf{x}_1^T \mathbf{K} \mathbf{x}_2 - \frac{k}{2} \mathbf{x}_2^T \mathbf{K}^T \mathbf{x}_1 + \mathbf{x}_2^T \mathbf{D}_\kappa \mathbf{x}_2 > 0 \quad (5.124)$$

when (5.122) holds. The stability condition for the vacuum case (5.118), is also a sufficient condition for any other isotropic material. It must be remarked that no assumptions on the homogeneity of the material were made.

The following conclusion can thus be drawn. The stability condition for an FDTD simulation based on a grid consisting of $n_x \times n_y$ Yee cells, with a perfectly conducting material $P_z = 0$ terminating the grid, and filled with any combination of isotropic materials, lossy or lossless, is:

$$\Delta_t < \frac{\Delta}{c_0 \sqrt{\cos^2 \left(\frac{\pi}{2(n_y+1)} \right) + \cos^2 \left(\frac{\pi}{2(n_x+1)} \right)}} \quad (5.125)$$

4.1.4 Vacuum case: boundary tangential O fields zero

In Section 4.3.1.5 of Chapter 2, a grid consisting of $n_x \times n_y$ Yee cells was presented. The tangential O_x and O_y field variables of this grid are zero at the boundary. The matrix \mathbf{K} in this case is (2.74):

$$\mathbf{K} = \begin{bmatrix} -\mathbf{I}_{n_x} \otimes \mathbf{W}_{n_y-1}^T \\ \mathbf{W}_{n_x-1}^T \otimes \mathbf{I}_{n_y} \end{bmatrix} \quad (5.126)$$

The singular values of \mathbf{K} are formed by the eigenvalues of the matrix

$$\mathbf{K}^T \mathbf{K} = \mathbf{I}_{n_x} \otimes (\mathbf{W}_{n_y-1} \mathbf{W}_{n_y-1}^T) + (\mathbf{W}_{n_x-1} \mathbf{W}_{n_x-1}^T) \otimes \mathbf{I}_{n_y} \quad (5.127)$$

These eigenvalues can be calculated in a similar fashion, by, first of all, taking into account that the eigenvalues of $\mathbf{W}_s \mathbf{W}_s^T$ are equal to the eigenvalues of $\mathbf{W}_s^T \mathbf{W}_s$ with one zero added. Remark that the size of $\mathbf{W}_s \mathbf{W}_s^T$ is $(s+1) \times (s+1)$, and the dimension of $\mathbf{W}_s^T \mathbf{W}_s$ is $s \times s$. Secondly by remarking that the eigenvalues of a certain matrix $\mathbf{A} + a\mathbf{I}$, are the eigenvalues of \mathbf{A} with a value a added. This is more specifically the case for $\mathbf{W}_{n_x-1} \mathbf{W}_{n_x-1}^T + a\mathbf{I}$.

Some calculations then lead to the following eigenvalues for $\mathbf{K}^T \mathbf{K}$

$$4 + 2 \cos \left(\frac{i\pi}{n_x} \right) + 2 \cos \left(\frac{j\pi}{n_y} \right) \quad \text{for } \begin{cases} i = 1, 2, \dots, n_x \\ j = 1, 2, \dots, n_y \end{cases} \quad (5.128)$$

The largest singular value of \mathbf{K} then is:

$$\sqrt{4 + 2 \cos \left(\frac{\pi}{n_x} \right) + 2 \cos \left(\frac{\pi}{n_y} \right)} \quad (5.129)$$

and the stability condition belonging to this grid is

$$\Delta_t < \frac{\Delta}{c_0 \sqrt{\cos^2\left(\frac{\pi}{2n_x}\right) + \cos^2\left(\frac{\pi}{2n_y}\right)}} \quad (5.130)$$

For very large problem domains, this stability condition approaches the Courant condition. As explained before, losses do not influence the stability. The influence of isotropic materials is identical as in the previous section and has a favourable effect on the stability condition.

4.1.5 Non reciprocal grid: Mur first order boundary conditions

In Section 2, the section discussing the stability related to the spatial discretization, it was illustrated that, when spatial reciprocity is lost, a system is potentially unstable. It is unlikely that, once the spatial discretization has created some unstable poles, the time discretized version of the problem will be stable. To demonstrate this, the problem of a grid containing a Mur first order absorbing boundary condition, discussed in Section 2.2.3 and shown in Fig. 5.8, containing the isotropic material, is reconsidered.

It was explained that the system is of the following form:

$$\begin{bmatrix} \mathbf{D} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}} \\ \dot{\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ \mathbf{p} \end{bmatrix} \quad (5.131)$$

Where \mathbf{D} is diagonal, but does not only contain values for θ_r , but also some non physical numbers related to the absorbing boundary condition. The matrix \mathbf{G}_{11} is no longer diagonal and $\mathbf{G}_{12} \neq -\mathbf{G}_{21}^T$. Time can be discretized as follows:

$$\frac{\mathbf{D}}{\Delta_t} (\mathbf{o}|^{n+1/2} - \mathbf{o}|^{n-1/2}) = -\frac{\mathbf{G}_{11}}{2} (\mathbf{o}|^{n+1/2} + \mathbf{o}|^{n-1/2}) - \mathbf{G}_{12}\mathbf{p}|^n \quad (5.132a)$$

$$\frac{\mathbf{I}}{\Delta_t} (\mathbf{p}|^{n+1} - \mathbf{p}|^n) = -\mathbf{G}_{21}\mathbf{o}|^{n+1/2} \quad (5.132b)$$

or in matrix form

$$\begin{bmatrix} \mathbf{D} + \frac{\Delta_t}{2}\mathbf{G}_{11} & 0 \\ \Delta_t\mathbf{G}_{21} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n+1/2} \\ \mathbf{p}|^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{D} - \frac{\Delta_t}{2}\mathbf{G}_{11} & -\Delta_t\mathbf{G}_{12} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{o}|^{n-1/2} \\ \mathbf{p}|^n \end{bmatrix} \quad (5.133)$$

The largest eigenvalue found, in absolute value, for the iteration matrix

$$\begin{bmatrix} \mathbf{D} + \frac{\Delta_t}{2}\mathbf{G}_{11} & 0 \\ \Delta_t\mathbf{G}_{21} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{D} - \frac{\Delta_t}{2}\mathbf{G}_{11} & -\Delta_t\mathbf{G}_{12} \\ 0 & \mathbf{I} \end{bmatrix} \quad (5.134)$$

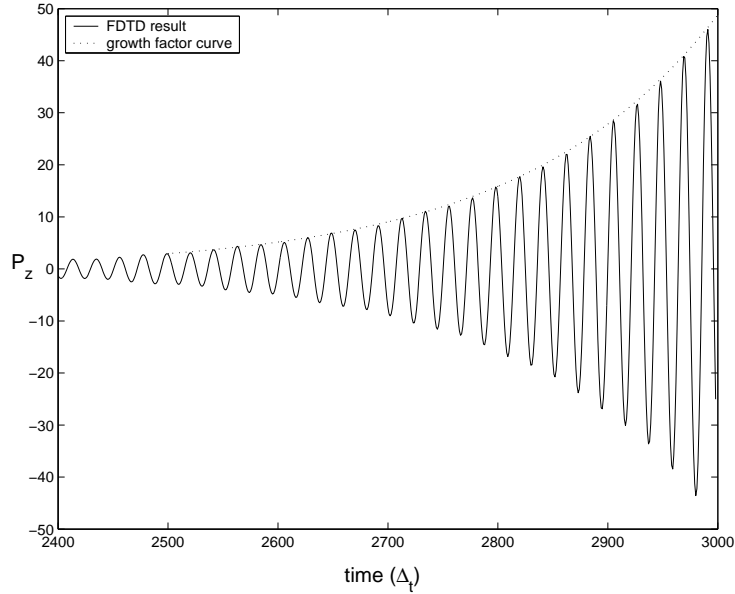


Figure 5.11: Simulated FDTD result for $k = 0.5$ and a curve predicting, based on the largest eigenvalue of the system, how the curve will grow.

was, for $k = 0.5$, $|\mu| = 1.0056386$. The largest eigenvalue, in absolute value, can be seen as the growth factor of the spurious result. To show this, a simulation, was performed. The time result indeed proved to be unstable, in Fig. 5.11 a part of the result was shown, $2400\Delta_t \leq t \leq 3000\Delta_t$. Also a curve starting with the value at $t = 2500\Delta_t$, and with a the growth factor, 1.0056386, was plotted. It can be observed that the rise of the FDTD result is very well described by this growth factor.

In Fig. 5.12, the largest eigenvalue pertaining to the problem presented in Fig. 5.8, as a function of k , is shown. The absolute value of the largest eigenvalue corresponds to the growth factor of the spurious solution. It can be seen that this factor is always larger than one. The problem is thus unconditionally unstable. Around the Courant condition $k = 1/\sqrt{2}$, the curve starts to rise very fast. It indicates that the FDTD stability condition or the Courant limit is still meaningful in this case. Choosing k larger than the Courant condition, makes the simulations useless right from the start.

The time step has an influence on how fast the spurious results grow. An interesting question in this context is to determine the time step that ensures that the spurious result has grown the least for a specific amount of simulated time. The time step that delays the late-time instability can be

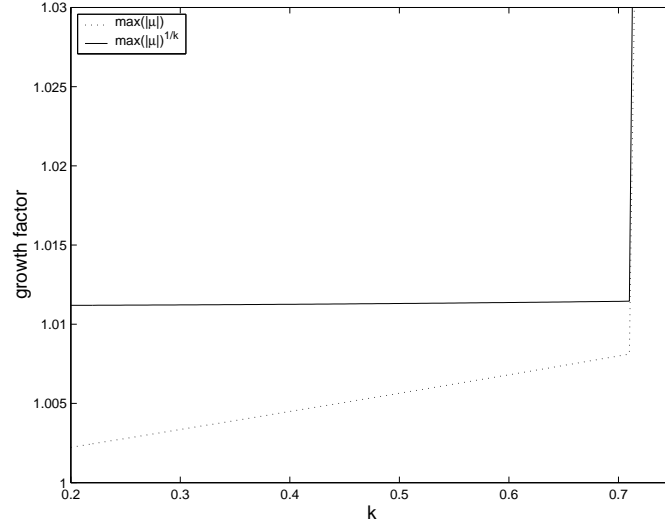


Figure 5.12: The extent of instability as a function of $k = c_0 \Delta_t / \Delta$ for the problem illustrated by Fig. 5.8. The dotted line indicates the growth factor per time step. The solid line shows this growth factor per unit of time.

determined by examining the growth factor per unit of time. This growth factor per unit of time is proportional to

$$(\max(|\mu(k)|))^{1/k} \quad (5.135)$$

since k is proportional to the time step, Δ_t . In Fig. 5.12, it can be observed that the growth factor per unit of time is fairly constant as long as $k < 1/\sqrt{2}$. Hence, since in this example the late-time instability cannot be pushed away, there is no reason to choose the time step smaller than proposed by the Courant condition.

4.2 ADI-FDTD method

In Section 4 of Chapter 3, the alternating-direction implicit FDTD (ADI-FDTD) method was introduced. There it was shown that the time stepping algorithm for the lossless case can be described by (3.41)

$$\left(\mathbf{D} + \frac{k}{2}\mathbf{B}\right)\mathbf{x}|^{n+1/2} = \left(\mathbf{D} - \frac{k}{2}\mathbf{A}\right)\mathbf{x}|^n \quad (5.136a)$$

$$\left(\mathbf{D} + \frac{k}{2}\mathbf{A}\right)\mathbf{x}|^{n+1} = \left(\mathbf{D} - \frac{k}{2}\mathbf{B}\right)\mathbf{x}|^{n+1/2} \quad (5.136b)$$

where the matrices $\mathbf{A} = -\mathbf{A}^T$ and $\mathbf{B} = -\mathbf{B}^T$ are skew symmetric and \mathbf{D} is diagonal, containing the relative constitutive parameters of the different field variables, see (3.38). The iteration matrix relating $\mathbf{x}|^{n+1}$ and $\mathbf{x}|^n$ is then:

$$\left(\mathbf{D} + \frac{k}{2}\mathbf{A}\right)^{-1} \left(\mathbf{D} - \frac{k}{2}\mathbf{B}\right) \left(\mathbf{D} + \frac{k}{2}\mathbf{B}\right)^{-1} \left(\mathbf{D} - \frac{k}{2}\mathbf{A}\right) \quad (5.137)$$

by introducing

$$\mathbf{D} = \mathbf{D}^{1/2} \mathbf{D}^{1/2} \quad (5.138)$$

this iteration matrix is similar to

$$\left(\mathbf{I} + \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right)^{-1} \left(\mathbf{I} - \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{B}\mathbf{D}^{-1/2}\right) \left(\mathbf{I} + \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{B}\mathbf{D}^{-1/2}\right)^{-1} \left(\mathbf{I} - \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right) \quad (5.139)$$

and also similar to

$$\left(\mathbf{I} - \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right) \left(\mathbf{I} + \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right)^{-1} \left(\mathbf{I} - \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{B}\mathbf{D}^{-1/2}\right) \left(\mathbf{I} + \frac{k}{2}\mathbf{D}^{-1/2}\mathbf{B}\mathbf{D}^{-1/2}\right)^{-1} \quad (5.140)$$

This matrix is the product of two matrices of the form:

$$\mathbf{Q} = (\mathbf{I} - \mathbf{F}) (\mathbf{I} + \mathbf{F})^{-1} \quad (5.141)$$

where $\mathbf{F} = -\mathbf{F}^T$ is skew symmetric. A matrix \mathbf{Q} of this form is an orthogonal matrix, $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. To show this, the eigenvalue decomposition of $\mathbf{F} = \mathbf{P}^{-1}\mathbf{\Lambda}\mathbf{P}$ is required:

$$\mathbf{Q}^T\mathbf{Q} = (\mathbf{I} + \mathbf{F})^{-T} (\mathbf{I} - \mathbf{F})^T (\mathbf{I} - \mathbf{F}) (\mathbf{I} + \mathbf{F})^{-1} \quad (5.142)$$

$$= (\mathbf{I} - \mathbf{F})^{-1} (\mathbf{I} + \mathbf{F}) (\mathbf{I} - \mathbf{F}) (\mathbf{I} + \mathbf{F})^{-1} \quad (5.143)$$

$$= \mathbf{P}^{-1} (\mathbf{I} - \mathbf{\Lambda})^{-1} (\mathbf{I} + \mathbf{\Lambda}) (\mathbf{I} - \mathbf{\Lambda}) (\mathbf{I} + \mathbf{\Lambda})^{-1} \mathbf{P} \quad (5.144)$$

$$= \mathbf{P}^{-1} (\mathbf{I} - \mathbf{\Lambda})^{-1} (\mathbf{I} - \mathbf{\Lambda}) (\mathbf{I} + \mathbf{\Lambda}) (\mathbf{I} + \mathbf{\Lambda})^{-1} \mathbf{P} \quad (5.145)$$

$$= \mathbf{I} \quad (5.146)$$

Along with this, use was made of the commuting nature of diagonal matrices, more specifically for $(\mathbf{I} + \mathbf{\Lambda})$ and $(\mathbf{I} - \mathbf{\Lambda})$.

In this way it is shown that the iteration matrix (5.137), is similar to the product of two orthogonal matrices. The product of two orthogonal matrices, say \mathbf{Q}_1 and \mathbf{Q}_2 , is also an orthogonal matrix:

$$\mathbf{Q}_1\mathbf{Q}_2 (\mathbf{Q}_1\mathbf{Q}_2)^T = \mathbf{Q}_1\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{Q}_1^T \quad (5.147)$$

$$= \mathbf{Q}_1\mathbf{Q}_1^T \quad (5.148)$$

$$= \mathbf{I} \quad (5.149)$$

And since the eigenvalues of an orthogonal matrix are all 1 in absolute value, it has been shown that the ADI-FDTD method is unconditionally stable.

4.3 Subdomain FDTD method

4.3.1 The subdomain FDTD method: the reciprocal case

First of all it is noted that, in the matrix update equation, (3.26) and (3.28), for a subdomain, the iteration matrix

$$\left(\mathbf{C} + \frac{\Delta_t}{2}\mathbf{G}\right)^{-1} \left(\mathbf{C} - \frac{\Delta_t}{2}\mathbf{G}\right) \quad (5.150)$$

describing how the internal field variables are updated, is an orthogonal matrix in the lossless case, see Section 4.2 on the stability of the ADI-FDTD method. When losses are present, the real part of the eigenvalues of the matrix

$$-\frac{\Delta_t}{2}\mathbf{C}^{-1}\mathbf{G} \quad (5.151)$$

determines the stability. When $(\mathbf{G} + \mathbf{G}^T) \geq 0$ and $\mathbf{C} > 0$, the eigenvalues of (5.150) are, in absolute value, smaller than or equal to 1. Therefore it can be concluded that this update equation, considered alone, is stable.

However, the subdomain update equations are part of a larger system. The stability is determined by the eigenvalues of the entire system. We start by considering a first subdomain, called subsystem 1:

$$\left\{ \begin{array}{l} \begin{bmatrix} \mathbf{D}_{\theta,1} & 0 \\ 0 & \mathbf{D}_{\kappa,1} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{o}}_1 \\ \dot{\mathbf{p}}_1 \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{\sigma,1} & \frac{c_0}{\Delta} \mathbf{K}_1 \\ -\frac{c_0}{\Delta} \mathbf{K}_1^T & \mathbf{D}_{\sigma^*,1} \end{bmatrix} \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{p}_1 \end{bmatrix} + \mathbf{B}_1 \mathbf{u}_1 \\ \mathbf{y}_1 = \mathbf{L}_1^T \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{p}_1 \end{bmatrix} \end{array} \right. \quad (5.152)$$

The subdomain is of the *O*-type, and is time discretized as in (3.26):

$$\left\{ \begin{array}{l} \begin{bmatrix} \mathbf{D}_{\theta,1} + \frac{\Delta_t}{2}\mathbf{D}_{\sigma,1} & \frac{k}{2}\mathbf{K}_1 \\ -\frac{k}{2}\mathbf{K}_1^T & \mathbf{D}_{\kappa,1} + \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*,1} \end{bmatrix} \begin{bmatrix} \mathbf{o}_1|^{n+1/2} \\ \mathbf{p}_1|^{n+1/2} \end{bmatrix} \\ = \begin{bmatrix} \mathbf{D}_{\theta,1} - \frac{\Delta_t}{2}\mathbf{D}_{\sigma,1} & -\frac{k}{2}\mathbf{K}_1 \\ \frac{k}{2}\mathbf{K}_1^T & \mathbf{D}_{\kappa,1} - \frac{\Delta_t}{2}\mathbf{D}_{\sigma^*,1} \end{bmatrix} \begin{bmatrix} \mathbf{o}_1|^{n-1/2} \\ \mathbf{p}_1|^{n-1/2} \end{bmatrix} + \Delta_t \mathbf{B}_1 \mathbf{u}_1|^{n-1/2} \\ \mathbf{y}_1|^{n+1/2} = \mathbf{L}_1^T \begin{bmatrix} \mathbf{o}_1|^{n+1/2} \\ \mathbf{p}_1|^{n+1/2} \end{bmatrix} \end{array} \right. \quad (5.153)$$

The rest of the simulation domain, called subsystem 2, is treated in the

normal FDTD way. In matrix form this is

$$\left\{ \begin{array}{l} \left[\begin{array}{cc} \mathbf{D}_{\theta,2} + \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & 0 \\ -k \mathbf{K}_2^T & \mathbf{D}_{\kappa,2} + \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{array} \right] \begin{bmatrix} \mathbf{o}_2|^{n+1/2} \\ \mathbf{p}_2|^{n+1} \end{bmatrix} \\ = \left[\begin{array}{cc} \mathbf{D}_{\theta,2} - \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & -k \mathbf{K}_2 \\ 0 & \mathbf{D}_{\kappa,2} - \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{array} \right] \begin{bmatrix} \mathbf{o}_2|^{n-1/2} \\ \mathbf{p}_2|^{n-1/2} \end{bmatrix} + \Delta t \mathbf{B}_2 \mathbf{u}_2|^{n+1/2} \\ \mathbf{y}_2|^{n-1/2} = \mathbf{L}_2^T \begin{bmatrix} \mathbf{o}_2|^{n-1/2} \\ \mathbf{p}_2|^{n-1/2} \end{bmatrix} \end{array} \right. \quad (5.154)$$

Analogous to the continuous time case, $\mathbf{u}_1|^{n-1/2} = \mathbf{y}_2|^{n-1/2}$ and $\mathbf{u}_2|^{n+1/2} = \mathbf{y}_1|^{n+1/2}$. Since the grid is supposed to be reciprocal, the transpose of $\mathbf{B}_1 \mathbf{L}_2^T$, is the opposite of $\mathbf{B}_2 \mathbf{L}_1^T$. Furthermore, since the subdomain is an *O*-type subdomain we obtain, see (2.79):

$$\Delta t \mathbf{B}_1 \mathbf{L}_2^T = \begin{bmatrix} 0 & k \mathbf{M} \\ 0 & 0 \end{bmatrix} \quad (5.155a)$$

$$\Delta t \mathbf{B}_2 \mathbf{L}_1^T = \begin{bmatrix} 0 & 0 \\ -k \mathbf{M}^T & 0 \end{bmatrix} \quad (5.155b)$$

The combination of both time stepping algorithms then yields:

$$\begin{aligned} & \left[\begin{array}{cccc} \mathbf{D}_{\theta,1} + \frac{\Delta t}{2} \mathbf{D}_{\sigma,1} & \frac{k}{2} \mathbf{K}_1 & 0 & 0 \\ -\frac{k}{2} \mathbf{K}_1^T & \mathbf{D}_{\kappa,1} + \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,1} & 0 & 0 \\ 0 & 0 & \mathbf{D}_{\theta,2} + \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & 0 \\ k \mathbf{M}^T & 0 & -k \mathbf{K}_2^T & \mathbf{D}_{\kappa,2} + \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{array} \right] \begin{bmatrix} \mathbf{o}_1|^{n+1/2} \\ \mathbf{p}_1|^{n+1/2} \\ \mathbf{o}_2|^{n+1/2} \\ \mathbf{p}_2|^{n+1} \end{bmatrix} \\ = & \left[\begin{array}{cccc} \mathbf{D}_{\theta,1} - \frac{\Delta t}{2} \mathbf{D}_{\sigma,1} & -\frac{k}{2} \mathbf{K}_1 & 0 & k \mathbf{M} \\ \frac{k}{2} \mathbf{K}_1^T & \mathbf{D}_{\kappa,1} - \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,1} & 0 & 0 \\ 0 & 0 & \mathbf{D}_{\theta,2} - \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & -k \mathbf{K}_2 \\ 0 & 0 & 0 & \mathbf{D}_{\kappa,2} - \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{array} \right] \begin{bmatrix} \mathbf{o}_1|^{n-1/2} \\ \mathbf{p}_1|^{n-1/2} \\ \mathbf{o}_2|^{n-1/2} \\ \mathbf{p}_2|^{n-1/2} \end{bmatrix} \end{aligned} \quad (5.156)$$

This can, just as in the standard FDTD case, be written as

$$(\mathbf{E} + \mathbf{F}) \mathbf{x}|^{n+1} = (\mathbf{E} - \mathbf{F}) \mathbf{x}|^n \quad (5.157)$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{D}_{\theta,1} & 0 & 0 & \frac{k}{2} \mathbf{M} \\ 0 & \mathbf{D}_{\kappa,1} & 0 & 0 \\ 0 & 0 & \mathbf{D}_{\theta,2} & -\frac{k}{2} \mathbf{K}_2 \\ \frac{k}{2} \mathbf{M}^T & 0 & -\frac{k}{2} \mathbf{K}_2^T & \mathbf{D}_{\kappa,2} \end{bmatrix} \quad (5.158a)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\Delta t}{2} \mathbf{D}_{\sigma,1} & \frac{k}{2} \mathbf{K}_1 & 0 & -\frac{k}{2} \mathbf{M} \\ -\frac{k}{2} \mathbf{K}_1^T & \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,1} & 0 & 0 \\ 0 & 0 & \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & \frac{k}{2} \mathbf{K}_2 \\ \frac{k}{2} \mathbf{M}^T & 0 & -\frac{k}{2} \mathbf{K}_2^T & \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{bmatrix} \quad (5.158b)$$

$$\mathbf{x}|^n = \begin{bmatrix} \mathbf{o}_1|^{n-1/2} \\ \mathbf{p}_1|^{n-1/2} \\ \mathbf{o}_2|^{n-1/2} \\ \mathbf{p}_2|^n \end{bmatrix} \quad (5.158c)$$

Following the same reasoning as in Section 4.1.1, it can be shown that when the symmetric part of \mathbf{F} is positive semidefinite, $(\mathbf{F} + \mathbf{F}^T) \geq 0$, the stability of the time stepping algorithm depends on \mathbf{E} , and on whether this matrix is positive definite. It is clear that $(\mathbf{F} + \mathbf{F}^T)$ is diagonal and proportional to the losses for each field variable, and consequently positive semidefinite. The positive definiteness of \mathbf{E} cannot exactly be determined, but is related to the stability of FDTD simulations. To be more precise, matrix \mathbf{E} does no longer depend on the internal field variables of the subdomain. Only the field variables of subsystem 2 and the output field variables of subsystem 1 determine the instability. The stability condition is identical to the stability of a standard FDTD simulation where all the internal field variables of the subdomain, this is not including the output field variables of the subdomain, are considered to be a perfectly conducting material. In Fig. 5.13 this is illustrated. Cut C indicates the location of an O -type subdomain. The stability of the simulation is identical to a standard FDTD simulation where all the subdomain internal field variables are omitted. Or equivalently an FDTD simulation, where all the omitted field variables of Fig. 5.13, form a perfect conducting material.

When the subdomain is replaced by some reduced model, the same reasoning can be repeated, but the matrices \mathbf{E} and \mathbf{F} change to:

$$\mathbf{E} = \begin{bmatrix} \mathbf{U}^T \mathbf{C}_1 \mathbf{U} & 0 & \frac{k}{2} \mathbf{U}_o^T \mathbf{M} \\ 0 & \mathbf{D}_{\theta,2} & -\frac{k}{2} \mathbf{K}_2 \\ \frac{k}{2} \mathbf{M}^T \mathbf{U}_o & -\frac{k}{2} \mathbf{K}_2^T & \mathbf{D}_{\kappa,2} \end{bmatrix} \quad (5.159a)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\Delta t}{2} \mathbf{U}^T \mathbf{G}_1 \mathbf{U} & 0 & -\frac{k}{2} \mathbf{U}_o^T \mathbf{M} \\ 0 & \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & \frac{k}{2} \mathbf{K}_2 \\ \frac{k}{2} \mathbf{M}^T \mathbf{U}_o & -\frac{k}{2} \mathbf{K}_2^T & \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{bmatrix} \quad (5.159b)$$

$$\mathbf{x}|^n = \begin{bmatrix} \mathbf{z}_1|^{n-1/2} \\ \mathbf{o}_2|^{n-1/2} \\ \mathbf{p}_2|^{n+1} \end{bmatrix} \quad (5.159c)$$

where, $\mathbf{U}^T = [\mathbf{U}_o^T \quad \mathbf{U}_p^T]^T$, belonging to the \mathbf{o}_1 and \mathbf{p}_1 field variables respec-

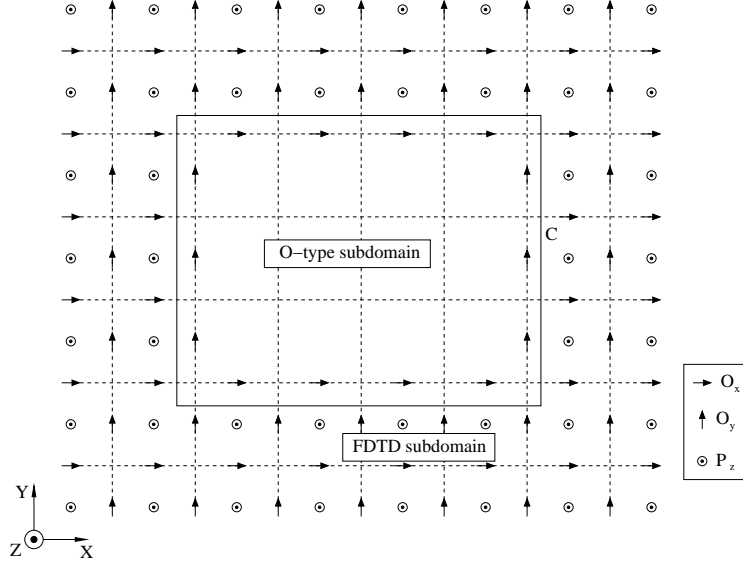


Figure 5.13: Cut C denotes an O -type subdomain, where spatial reciprocity is maintained. The surrounding grid is treated by the standard FDTD equations. Stability of the simulation only depends on the field variables shown.

tively was introduced, and where

$$\mathbf{U}^T \mathbf{C}_1 \mathbf{U} = \begin{bmatrix} \mathbf{U}_o^T & \mathbf{U}_p^T \end{bmatrix} \begin{bmatrix} \mathbf{D}_{\theta,1} & 0 \\ 0 & \mathbf{D}_{\kappa,1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_o \\ \mathbf{U}_p \end{bmatrix} \quad (5.160a)$$

$$= \mathbf{U}_o^T \mathbf{D}_{\theta,1} \mathbf{U}_o + \mathbf{U}_p^T \mathbf{D}_{\kappa,1} \mathbf{U}_p \quad (5.160b)$$

and

$$\frac{\Delta t}{2} \mathbf{U}^T \mathbf{G}_1 \mathbf{U} = \begin{bmatrix} \mathbf{U}_o^T & \mathbf{U}_p^T \end{bmatrix} \begin{bmatrix} \frac{\Delta t}{2} \mathbf{D}_{\sigma,1} & \frac{k}{2} \mathbf{K}_1 \\ -\frac{k}{2} \mathbf{K}_1^T & \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_o \\ \mathbf{U}_p \end{bmatrix} \quad (5.161a)$$

$$= \frac{\Delta t}{2} \mathbf{U}_o^T \mathbf{D}_{\sigma,1} \mathbf{U}_o + \frac{k}{2} \mathbf{U}_o^T \mathbf{K}_1 \mathbf{U}_p - \frac{k}{2} \mathbf{U}_p^T \mathbf{K}_1^T \mathbf{U}_o + \frac{\Delta t}{2} \mathbf{U}_p^T \mathbf{D}_{\sigma^*,1} \mathbf{U}_p \quad (5.161b)$$

Showing again that the symmetric part of \mathbf{F} is positive semidefinite, $(\mathbf{F} + \mathbf{F}^T) \geq 0$, transforms the question of stability, to the question when \mathbf{E} is positive definite. Determining a bound for k , for which $\mathbf{E} > 0$, is not easy. The link between the stability of the time stepping algorithm and the stability of a related FDTD problem is no longer possible, since the field variables of the subdomain, $\begin{bmatrix} \mathbf{o}_1^T & \mathbf{p}_1^T \end{bmatrix}$ have been replaced by some new variables \mathbf{z}_1 . However, the analysis demonstrates that there must be

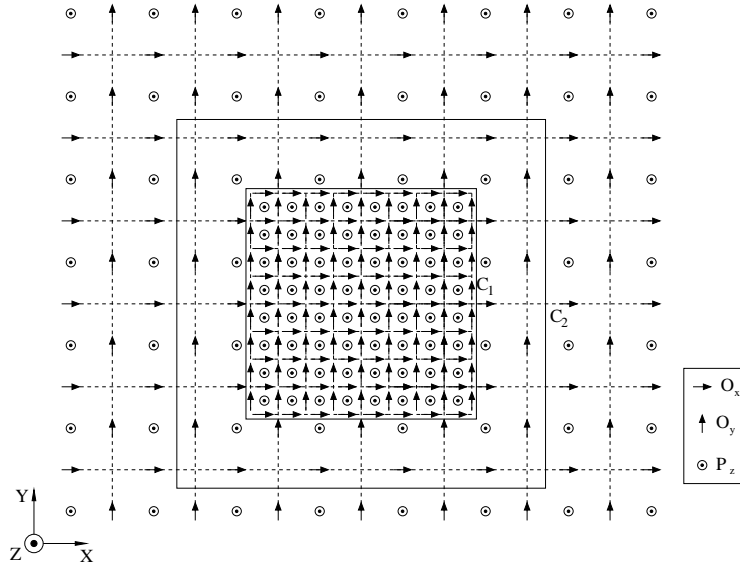


Figure 5.14: A FDTD grid including one subgridded subdomain, two possible cuts dividing the system into two systems are shown: cut C_1 , at the boundary of fine grid and coarse grid, cut C_2 , the smallest cut including the non-reciprocal region.

a bound, $k < k_0$, for which the time stepping algorithm is stable. This can be seen by noting that for small k , the different diagonal blocks dominate the symmetric matrix, and that each of these blocks is positive definite separately.

The analysis performed up to now was executed making use of some O -type subdomain. It can easily be repeated for a P -type subdomain.

4.3.2 The subdomain FDTD method: treating the non-reciprocal case

Unfortunately, a similar analysis as in previous section, does not guarantee stability for non-reciprocal grids. This is the case, when subgridded subdomains are present, as shown in Chapter 2. In Fig. 5.14, a typical FDTD grid is shown including one subgridded subdomain. Two cuts separate subsystem 1 from subsystem 2.

Cut C_1 separates the fine grid field variables from the coarse grid field variables, cut C_2 is introduced just outside cut C_1 and includes the non-reciprocal part of the grid. In Fig. 5.14, both subdomains are O -type sub-

domains. The matrices \mathbf{E} and \mathbf{F} are as follows:

$$\mathbf{E} = \begin{bmatrix} \mathbf{U}^T \mathbf{C}_1 \mathbf{U} & 0 & \frac{k}{2} \mathbf{U}_o^T \mathbf{M}_\alpha \\ 0 & \mathbf{D}_{\theta,2} & -\frac{k}{2} \mathbf{K}_2 \\ \frac{k}{2} \mathbf{M}_\beta^T \mathbf{U}_o & -\frac{k}{2} \mathbf{K}_2^T & \mathbf{D}_{\kappa,2} \end{bmatrix} \quad (5.162a)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\Delta t}{2} \mathbf{U}^T \mathbf{G}_1 \mathbf{U} & 0 & -\frac{k}{2} \mathbf{U}_o^T \mathbf{M}_\alpha \\ 0 & \frac{\Delta t}{2} \mathbf{D}_{\sigma,2} & \frac{k}{2} \mathbf{K}_2 \\ \frac{k}{2} \mathbf{M}_\beta^T \mathbf{U}_o & -\frac{k}{2} \mathbf{K}_2^T & \frac{\Delta t}{2} \mathbf{D}_{\sigma^*,2} \end{bmatrix} \quad (5.162b)$$

The grid is not reciprocal and therefore it is not possible to prove that $(\mathbf{F} + \mathbf{F}^T)$ is positive semidefinite. For cut C_1 , this is caused by $\mathbf{M}_\alpha \neq \mathbf{M}_\beta$, for cut C_2 , the cause is that matrix $\mathbf{U}^T (\mathbf{G}_1 + \mathbf{G}_1^T) \mathbf{U}$ is no longer positive semidefinite. Note that for both C_1 and C_2 , $\mathbf{U}^T \mathbf{C}_1 \mathbf{U}$ remains positive definite. It is thus possible, when C_2 separates the subdomain from the surrounding grid, to stabilize the entire system by approximating $\mathbf{U} \mathbf{G}_1 \mathbf{U}$ by some matrix \mathbf{P} for which $(\mathbf{P} + \mathbf{P}^T) > 0$. Thanks to the reduced order modeling, the matrix $\mathbf{U} \mathbf{G}_1 \mathbf{U}$ is of a limited size. This implies that the calculation time to construct a matrix \mathbf{P} is limited.

Just as before, there is no limit that prevents repeating this analysis for P -type subdomains. Even so, when several subdomains are defined inside a standard FDTD grid, the same conclusions can be drawn.

4.3.3 The generalized subdomain FDTD method

The generalized subdomain FDTD method divides the grid into two types of subdomains, in a chessboard kind of way. The first type of subdomains is time discretized around whole time steps, $t = n\Delta t$, (5.153):

$$\begin{cases} \left(\mathbf{C}_1 + \frac{\Delta t}{2} \mathbf{G}_1 \right) \mathbf{x}_1|^{n+1/2} = \left(\mathbf{C}_1 - \frac{\Delta t}{2} \mathbf{G}_1 \right) \mathbf{x}_1|^{n-1/2} + \Delta t \mathbf{B}_1 \mathbf{u}_1|^{n+1/2} \\ \mathbf{y}_1|^{n+1/2} = \mathbf{L}_1^T \mathbf{x}_1|^{n+1/2} \end{cases} \quad (5.163)$$

The second type of subdomains is time discretized around half time steps, $t = (n + 1/2)\Delta t$:

$$\begin{cases} \left(\mathbf{C}_2 + \frac{\Delta t}{2} \mathbf{G}_2 \right) \mathbf{x}_2|^{n+1} = \left(\mathbf{C}_2 - \frac{\Delta t}{2} \mathbf{G}_2 \right) \mathbf{x}_2|^{n+1/2} + \Delta t \mathbf{B}_2 \mathbf{u}_2|^{n+1/2} \\ \mathbf{y}_2|^{n+1} = \mathbf{L}_2^T \mathbf{x}_2|^{n+1} \end{cases} \quad (5.164)$$

In order to avoid a too complicated analysis, a system consisting of two subdomains is considered. The system time iteration scheme can then be summarized by

$$(\mathbf{E} + \mathbf{F}) \mathbf{x}|^{n+1} = (\mathbf{E} - \mathbf{F}) \mathbf{x}|^n \quad (5.165)$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{C}_1 & \frac{\Delta t}{2} \mathbf{B}_2 \mathbf{L}_1^T \\ \frac{\Delta t}{2} \mathbf{B}_1 \mathbf{L}_2^T & \mathbf{C}_2 \end{bmatrix} \quad (5.166a)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\Delta t}{2} \mathbf{G}_1 & -\frac{\Delta t}{2} \mathbf{B}_2 \mathbf{L}_1^T \\ -\frac{\Delta t}{2} \mathbf{B}_1 \mathbf{L}_2^T & \frac{\Delta t}{2} \mathbf{G}_2 \end{bmatrix} \quad (5.166b)$$

$$\mathbf{x}|^n = \begin{bmatrix} \mathbf{x}_1|^{n-1/2} \\ \mathbf{x}_2|^n \end{bmatrix} \quad (5.166c)$$

It can be shown that for a reciprocal system, the stability is only determined by the positive definiteness of \mathbf{E} . The only part disturbing the diagonal character of \mathbf{C} are the entries due to $\mathbf{B}_1 \mathbf{L}_2^T = -(\mathbf{B}_2 \mathbf{L}_1^T)^T$. Only the input and output field variables of the subdomains determine the stability of the system.

This can be seen from a more physical point of view: inside each subdomain a numerical wave can propagate at each speed, due to the implicit nature of the update equations. However at the transition region between both subdomains, the wave can numerically only propagate over one space step in one time step. For large time steps, the propagated distance per time step would exceed the space step and the algorithm is not capable of following the wave propagation.

For reduced order subdomain models, the matrix \mathbf{E} transforms to:

$$\mathbf{E} = \begin{bmatrix} \mathbf{U}_1^T \mathbf{C}_1 \mathbf{U}_1 & \frac{\Delta t}{2} \mathbf{U}_2^T \mathbf{B}_2 \mathbf{L}_1^T \mathbf{U}_1 \\ \frac{\Delta t}{2} \mathbf{U}_1^T \mathbf{B}_1 \mathbf{L}_2^T \mathbf{U}_2 & \mathbf{U}_2^T \mathbf{C}_2 \mathbf{U}_2 \end{bmatrix} \quad (5.167)$$

and a stability bound still exists, since $(\mathbf{U}_2^T \mathbf{B}_2 \mathbf{L}_1^T \mathbf{U}_1)^T = -\mathbf{U}_1^T \mathbf{B}_1 \mathbf{L}_2^T \mathbf{U}_2$.

5 OVERVIEW

In this chapter stability was discussed. In each step of the different FDTD algorithms, a source of instability can be discovered: in the spatial discretization step, in the ROM step and in the time discretization step. Once something has gone wrong in a certain step, one cannot remove this source of instability in the subsequent steps.

In the spatial discretization step, poles can end up in the right half plane. However, when the grid is spatially reciprocal all the poles are guaranteed not to be located in the right half plane. Even for some special FDTD features, e.g. the thin wire subcell model, it was shown that the system could be transformed to the desired reciprocal form, guaranteeing stability. However, when the spatial reciprocity property is lost, it is no longer

possible to give any guarantee on the location of the poles. Subgridding is a technique used in the subdomain FDTD method that does not preserve spatial reciprocity. An example of a stable and an unstable system was given. To put things into perspective, the Mur first order ABC, which also does not preserve spatial reciprocity, was studied. An example leading to an unstable system was also given.

In the ROM step, it was shown that the matrices \mathbf{C} and $(\mathbf{G} + \mathbf{G}^T)$ need to be positive definite, otherwise the reduced system has unstable poles. It was also shown that in this way the reciprocal or non-reciprocal nature of the system is maintained.

In the time discretization step, depending on the way time discretization was performed, a maximum value for k , where $k = c_0 \Delta_t / \Delta$, resulting in a stable algorithm, could be derived. For the standard FDTD method, a stability condition for a finite grid was derived. Furthermore it was proven that this condition holds when isotropic materials and losses are present. For the ADI-FDTD method, the unconditional stability for a lossless finite grid was shown. For the subdomain FDTD method, when dealing with a spatially reciprocal grid, it was shown that the stability condition was related to the stability condition of a standard FDTD problem. For the subdomain FDTD method, based on a non-reciprocal grid, a suggestion was made on how to transform the problem to a stable one.

BIBLIOGRAPHY

- [1] A. Taflové and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 23, no. 8, pp. 623-630, Aug. 1975.
- [2] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 10, pp. 2003-2007, Oct. 1999.
- [3] F. Zheng, Z. Chen, and J. Zhang, "Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1550-1558, Sep. 2000.
- [4] A. Taflové, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 1995.

- [5] A. S. Householder, *The theory of matrices in numerical analysis*, Dover Publications, Inc, New York, 1964.
- [6] A. Taflove, K. R. Umashankar, B. Beker, F. Harfoush, and K. S. Yee, "Detailed FD-TD analysis of electromagnetic fields penetrating narrow slots and lapped joints in thick conducting screens," *IEEE Trans. Antennas Propagat.*, vol. 36, no. 2, pp. 247-257, Feb. 1988.
- [7] P. Thoma and T. Weiland, "A consistent subgridding scheme for the finite difference time domain method," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 9, no. 5, pp. 359-374, 1996.
- [8] K. M. Krishnaiah and C. J. Railton, "Passive equivalent circuit of FDTD: an application to subgridding," *Electronics Letters*, vol. 33, no. 15, pp. 1277-1278, July 1997.
- [9] K. M. Krishnaiah and C. J. Railton, "A stable subgridding algorithm and its application to eigenvalue problems," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 5, pp. 620-628, May 1999.
- [10] S. Wang, F. L. Teixeira, R. Lee, and J.-F. Lee, "Optimization of subgridding schemes for FDTD," *IEEE Microwave and Wireless Comp. Letters*, vol. 12, no. 6, pp. 223-225, June 2002.
- [11] C. J. Railton, I. J. Craddock, and J. B. Schneider, "Improved locally distorted CPFDTD algorithm with provable stability," *Electronics Letters*, vol. 31, no. 18, pp. 1585-1586, Aug. 1995.
- [12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.
- [13] S. Van den Berghe, *Object-oriented Electromagnetic Simulations with the Finite-Difference Time-Domain Method*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1999.
- [14] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time domain electromagnetic field equations," *IEEE Trans. Electromagnetic Compatibility*, vol. 23, no. 4, pp. 377-382, 1981.
- [15] J. P. Bérenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Computational Physics*, vol. 114, no. 1, pp. 185-200, 1994.

-
- [16] L. Knockaert and D. De Zutter, "Passive reduced order multiport modeling: The Padé-Laguerre, Krylov-Arnoldi-SVD connection," *AEU Int. J. Electron Commun.*, vol. 53, no. 5, pp. 254–260, 1999.
 - [17] J. A. Pereda, O. García, A. Vegas, and A. Prieto, "Numerical dispersion and stability analysis of the FDTD technique in lossy dielectrics," *IEEE Microwave and Guided Wave Letters*, vol. 8, no. 7, pp. 245–247, July 1998.
 - [18] R. F. Remis, "On the stability of the finite-difference time-domain method," *J. Computational Physics*, vol. 163, no. 1, pp. 249–261, Sep. 2000.
 - [19] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Clarendon Press, Oxford, 1985.
 - [20] R. F. Remis, "Global form of the 2D finite-difference maxwell system and its symmetry properties," *IEEE Antennas Propagat. Symp.*, vol. 3, pp. 624–627, June 2002.

CHAPTER 6

Numerical examples

1 INTRODUCTION

In this chapter a number of numerical examples will be studied. They will be of two possible classes: first of all, in Section 2 an example of the generalized subdomain FDTD method will be studied. Afterwards, in Section 3, some examples of the subdomain FDTD method will be analyzed. This corresponds to the chronological order in which both methods have appeared. First the generalized subdomain FDTD method was proposed in [1] and afterwards in [2] the subdomain FDTD method. The naming of both methods was only derived afterwards based on further theoretical understanding.

2 GENERALIZED SUBDOMAIN FDTD METHOD

The generalized subdomain FDTD method was presented in Section 3.4 of Chapter 3. A problem space is divided into subdomains. Once a ROM of the different subdomains is generated, they are updated in an alternating fashion, using equations (3.26) and (3.28). A parallel-plate waveguide problem, with a metallic object causing reflection (Fig. 6.1), was studied. Some of these results have been published in [1], [3].

This is a TE problem: $O_x \sim E_x$, $O_y \sim E_y$ and $P_z \sim H_z$. The reflection and transmission of a propagating TEM wave was calculated. The infinite waveguide was modelled by a 63 cm long waveguide terminated with Mur first order absorbing boundary conditions. The distance between the plates is 3.15 cm and the waveguide is filled with air. The TEM-mode is the only propagating mode, the first higher mode only propagates for $f > 4.76$ GHz. The number of cells is $n_x = 180$ and $n_y = 9$ and the size of the cells is

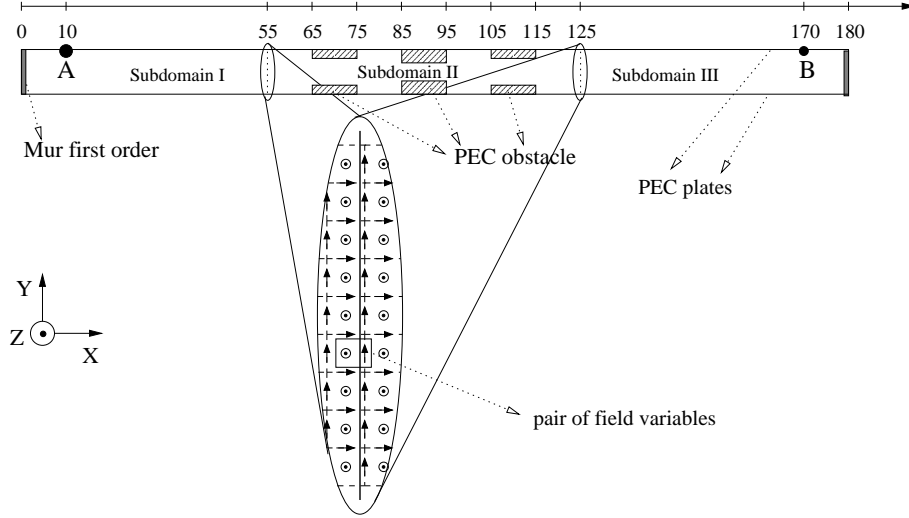


Figure 6.1: TEM simulation problem.

$\Delta = 3.5$ mm. The waveguide contains 3 metallic irises (Fig. 6.1), the first and third iris, between $65\Delta \geq x \geq 75\Delta$ and $105\Delta \geq x \geq 115\Delta$ respectively have a spacing of 5Δ . The second iris, between $85\Delta \geq x \geq 95\Delta$, left a smaller opening of 3Δ . The TEM-mode was excited by a current injected at $x = 10\Delta$ (location A in Fig. 6.1). The source was a Gaussian pulse.

The reflected field was also recorded at location A, the transmitted field at $x = 170\Delta$ (location B in Fig. 6.1). The fast Fourier transform (FFT) was used to calculate the frequency results. A 1-D FDTD simulation, with 361 variables, corresponding to the same waveguide with the irises removed, was used to calculate the incident pulse. This incident pulse was required to calculate the reflected pulse from the total pulse. The reference result was a standard FDTD simulation based on the same grid.

The computational domain was divided into three subdomains, by means of a cut at cell 55 and at cell 125, as shown on Fig. 6.1. Subdomain I contained 1430 field variables, subdomain II 1386 and subdomain III 1439. The cut was introduced in such a way that for each cut, the field variables just left of the cut or the right side subdomain boundary field variables are H_z , and the field variables just right of the cut or the left side subdomain boundary field variables are E_y . Since the problem domain is terminated by perfect electric conductors at the upper and the lower side, there are only contributions to \mathbf{u} and \mathbf{y} for each subdomain at the left and right sides of these domains. In this way the number of boundary field variables for

each subdomain has been limited.

Since we are dealing with a TEM problem, and by choosing the cut far enough from the irises, the only mode present at the cut is this TEM mode. As a result of this, it is not necessary to consider all input and output field variables at the subdomain boundary separately since they are, to a great extent, equal. Replacing at each cut the output field variables by their average, or equivalently the amplitude of the mode, the simulations can be performed faster, since each reduced model has dimensions proportional to the number of input field variables. The output and input field variables for each subdomain are then as follows:

- for subdomain I, one input and output variable at the right subdomain boundary, and furthermore one input variable for the source at location A, and one output for recording the field at location A.
- for subdomain II, one input and output variable at the left subdomain boundary and one input and output field variable at the right subdomain boundary
- for subdomain III, one input and output variable at the left subdomain boundary, and furthermore one output for the recorded field at location B, and one input for possible use as a source located at B.

In this way each subdomain has two inputs and two outputs: $p = 2$. Each simulation was based on reduced models with the same order of approximation $q = q_I = q_{II} = q_{III}$. Therefore the reduced models of each subdomain have the same dimension: each subdomain is described using $2q$ variables. The leapfrog timestepping algorithm is: at $t = n\Delta_t$, subdomain I and subdomain III are updated, and at $t = (n + 1/2)\Delta_t$ subdomain II is updated.

In Fig. 6.2, the reflection coefficient for $q = 5, 6, 7$ is compared to the reflection coefficient obtained from the FDTD simulation, and in Fig. 6.3, the same was done for $q = 8, 9, 10$. For each simulation the parameter $\alpha = 2\pi f_{\max}$, used in the ROM algorithm, was kept equal and set to $f_{\max} = 0.25$ GHz. The time step, for each simulation, was chosen in agreement to the Courant condition: $\Delta_t = 8.24 \cdot 10^{-12}$ s.

Both figures clearly show the results for the progressively improving models. The more complex the reduced models are, or equivalently, the higher q is chosen, the higher the frequency up to which the results hold. For $q = 5$, the frequency up to which results agree is over 0.75 GHz, for $q = 6$, this frequency is already 1.25 GHz and for $q = 7$, 1.5 GHz. For $q = 8, 9, 10$, the results keep improving, but not at the same pace.

The same observations can be made for the transmission coefficient. In Fig. 6.4 the transmission coefficient is shown for $q = 5, 6, 7$ together

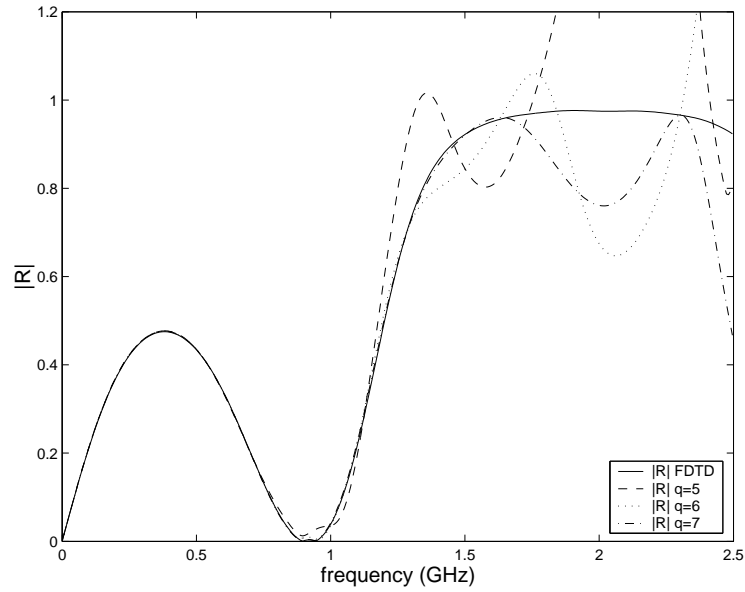


Figure 6.2: Reflection coefficient for $q = 5, 6, 7$ compared to the FDTD reference result.

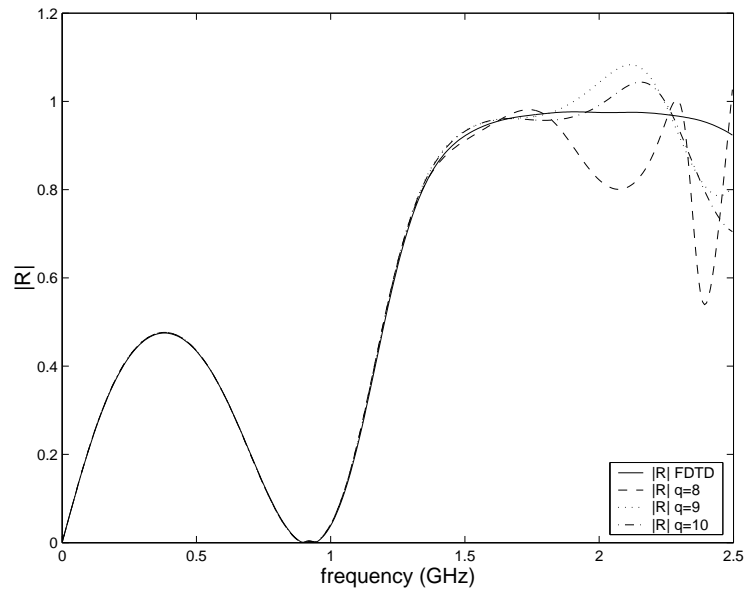


Figure 6.3: Reflection coefficient for $q = 8, 9, 10$ compared to the FDTD reference result.

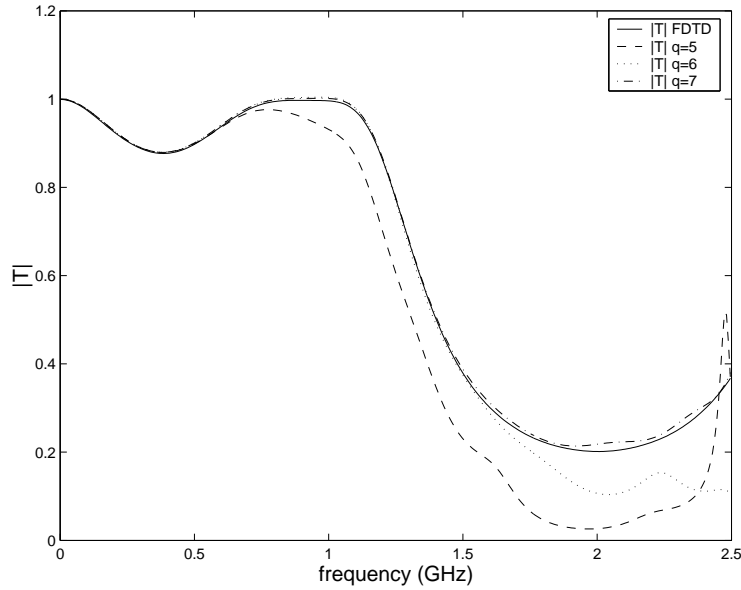


Figure 6.4: Transmission coefficient for $q = 5, 6, 7$ compared to the FDTD reference result.

with the FDTD generated result. The frequencies for which the results are accurate, are the same as for the reflection coefficient. In Fig. 6.5, the difference between the FDTD method and simulations with ROM's $q = 7, 8, 9, 10$, is shown, in dB. It shows that, generally speaking, for increasing q , the frequency for which a certain level of accuracy, say 40 dB, is crossed, increases.

To prove that, at each subdomain boundary, the input and output field variables can be replaced by a single input and output variable giving the amplitude of the TEM-mode, the difference between a full field simulation and the single mode based simulation is evaluated, see Fig. 6.6. Two results for $q = 7$ and $f_{\max} = 0.25$ GHz, are compared. The figure indicates that both results correspond very well in the frequency band, $f < 1.5$ GHz, where the subdomain models are reliable. The difference between the full field simulation, involving a reduced model for subdomain II of $18 \times 7 = 126$ variables, and the single mode based simulation, involving a reduced model for subdomain II of $2 \times 7 = 14$ variables, is at the most 40dB in that frequency band.

For the ROM algorithm a choice for α needs to be made. In [4],

$$\alpha = 2\pi f_{\max} \quad (6.1)$$

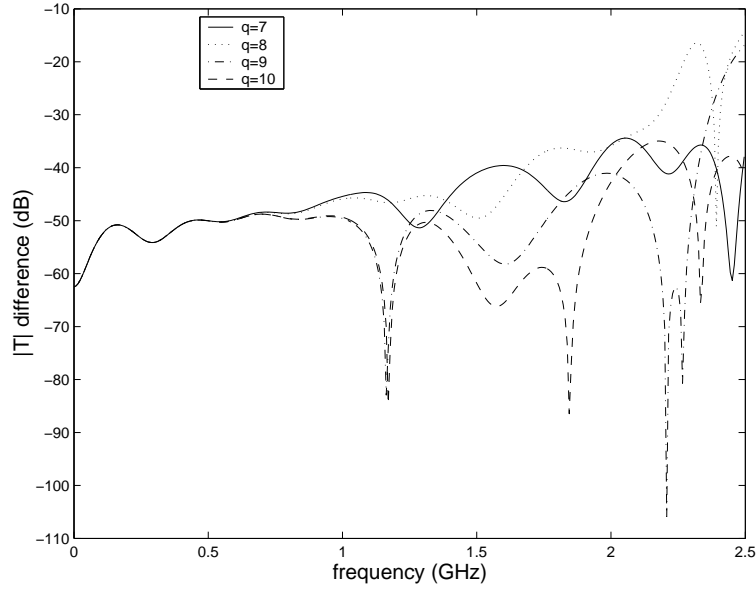


Figure 6.5: The difference between the transmission coefficient for $q = 7, 8, 9, 10$ and the FDTD transmission coefficient in dB.

with f_{\max} the frequency bandwidth of the system is proposed. The influence of this parameter is limited. In Fig. 6.7, for $q = 7$, the reflection coefficient is shown for three choices of f_{\max} : 0.125 GHz, 0.25 GHz and 0.375 GHz. It can be observed that for $f_{\max} = 0.125$ GHz, the results deteriorate, while no difference can be observed between $f_{\max} = 0.25$ GHz and 0.375 GHz. Other results for $f_{\max} = 0.5$ GHz, $f_{\max} = 0.625$ GHz and $f_{\max} = 0.75$ GHz did not show any difference with the case where $f_{\max} = 0.25$ GHz. Other tests also indicated that the choice of α only plays a minor role.

Up to now the time step for each simulation corresponded to the Courant condition associated with the space step $\Delta = 3.5$ mm: $\Delta_t = 8.24 \cdot 10^{-12}$ s. In Table 6.1, for $q = 5, \dots, 8$, the maximum time step, $\Delta_{t,\max}/\Delta_t$, for which stable results are obtained, as a multiple of $\Delta_t = 8.24 \cdot 10^{-12}$ s, was determined numerically. It can be observed that the time step can be chosen considerably larger, a factor four to eight, for the q -values concerned. It is interesting to note that the smaller the reduced model, smaller q , the higher the time step can be chosen. In the adjacent column

$$k_{\max} = \frac{c_0 \Delta_{t,\max}}{\Delta} \quad (6.2)$$

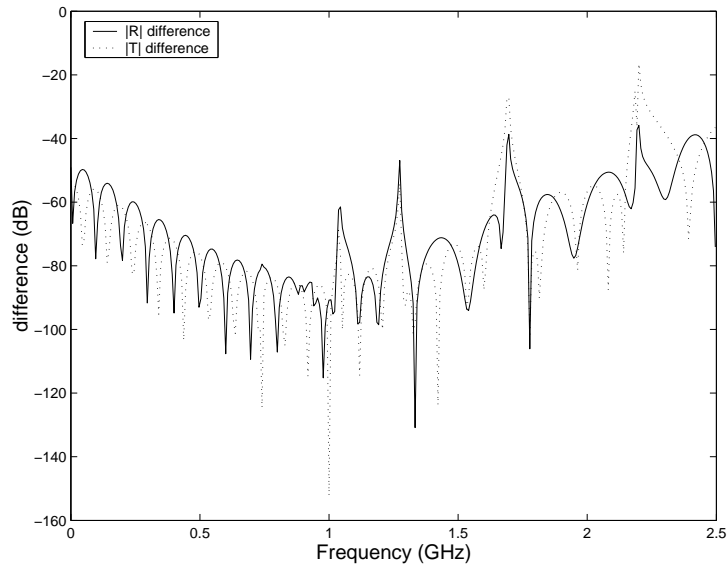


Figure 6.6: Difference for reflection and transmission between a full field simulation and a simulation where a single variable relates the subdomains, when $q = 7$ and $f_{\max} = 0.25$ GHz.

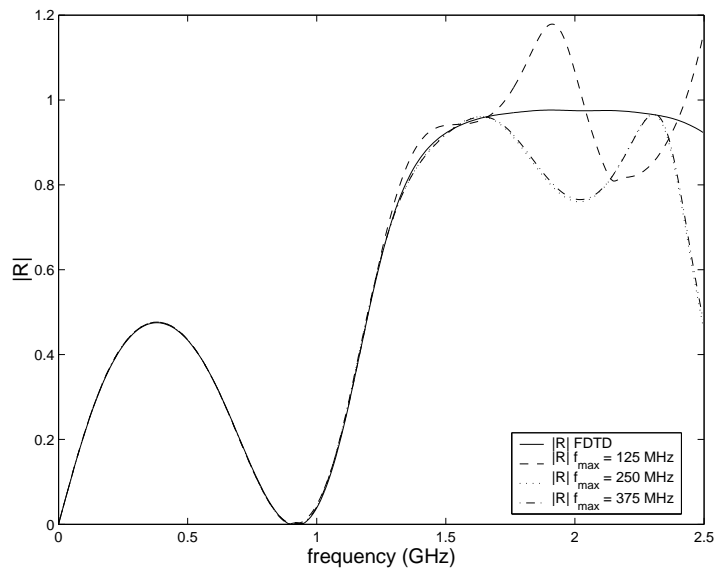


Figure 6.7: Reflection coefficient for $f_{\max} = 0.125$ GHz, $f_{\max} = 0.25$ GHz and $f_{\max} = 0.375$ GHz when $q = 7$.

| q | $\Delta_{t,\max}/\Delta_t$ | k_{\max} | σ_1 | $2/\sigma_1$ |
|-----|----------------------------|------------|------------|--------------|
| 5 | 8.34 | 5.89 | 0.331 | 6.04 |
| 6 | 6.18 | 4.36 | 0.460 | 4.35 |
| 7 | 5.01 | 3.54 | 0.566 | 3.53 |
| 8 | 4.64 | 3.27 | 0.602 | 3.32 |

Table 6.1: The maximum time step, derived from the subdomain input and output matrices (σ_1), and the maximum time step observed (k_{\max}).

is shown.

In Section 4.3.3 of Chapter 5, a stability relation for the generalized subdomain FDTD method with two subdomains was derived. It was noted that only the matrices relating input and output field variables of the subdomains, and more specific the largest singular value of $\tilde{\mathbf{B}}_1 \tilde{\mathbf{L}}^T$, determine the stability of the algorithm. Note first of all that, since Mur ABC's are used, we are not dealing with a reciprocal grid, and further that the problem actually encompasses three subdomains. In Table 6.1, the largest singular value, σ_1 , of $\frac{\Delta}{c_0} \mathbf{U}_1^T \mathbf{B}_1 \mathbf{L}_2^T \mathbf{U}_2$ is shown. Matrix \mathbf{B}_1 is the column vector related to the input variable coming from subdomain II and \mathbf{L}_2 is the column vector defining the output variable needed in subdomain I. Only the parts from \mathbf{B}_1 and \mathbf{L}_2 related to the cut between subdomain I and subdomain II are considered. In the last column the value $2/\sigma_1$ is shown.

The correspondence between k_{\max} and $2/\sigma_1$ is striking. This was expected since eq. (5.100), giving a relation between k and the largest singular value of the off-diagonal blocks of a matrix, is also useful here (5.167). The reason for the deviation, especially for $q = 5$ and $q = 8$, can be explained since for k_{\max} three subdomains were considered, whereas only two subdomains for σ_1 . The largest singular value for the unreduced version, (5.166) was found to be one, $\sigma_1 = 1$. This corresponds to the stability limit of the 1-D FDTD algorithm. This is a logical result since each subdomain in itself is unconditionally stable and only the transition from one subdomain to the other at each cut determines stability. More specifically at each cut pairs of field variables are related by means of an explicit formula: an H_z field at the left hand side of the cut and its neighbour an E_y field at the right hand side of the cut, see Fig. 6.1. Each of these pairs acts as an individual 1-D FDTD grid, a half cell in size.

In Fig. 6.8, a closer look was taken at the influence an increased time step has on the accuracy. For $q = 7$, the time step was multiplied by four, resulting in $\Delta_t = 3.296 \cdot 10^{-11}$ s. The previous result for $q = 7$, with the old time step and the reference FDTD result was added for comparison.

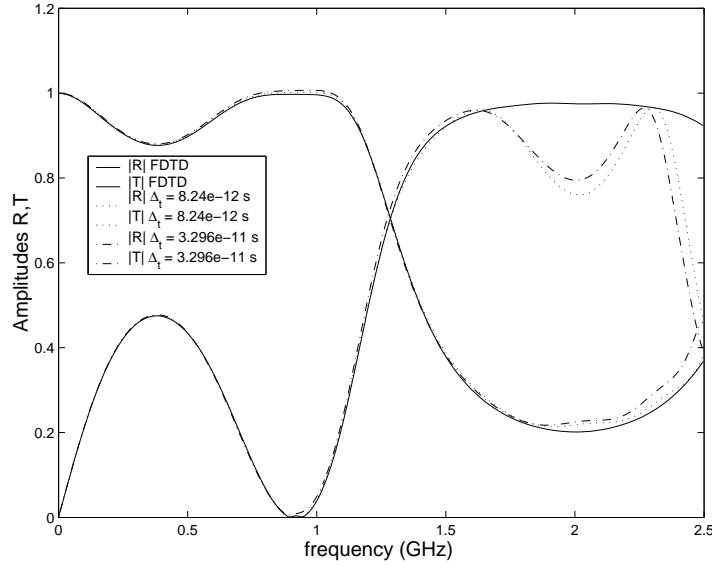


Figure 6.8: Difference for reflection and transmission between a full field simulation and a simulation where a single variable connects the subdomains, when $q = 7$ and $f_{\max} = 0.25$ GHz.

The result has worsened slightly, the deviation between both ROM based results increases as the frequency increases.

Thanks to this new approach, the computational savings were considerable. Considering the matrix update equation

$$\begin{cases} \mathbf{z}^{n+1} = \mathbf{\Lambda} \mathbf{z}^n + \mathbf{F} \mathbf{u}^{n+1/2} \\ \mathbf{y}^{n+1} = \mathbf{E}^T \mathbf{z}^{n+1} \end{cases} \quad (6.3)$$

where, for the sake of clarity \mathbf{E} and \mathbf{F} are not related to matrices with the same name used in the stability analysis. Each subdomain has only $2q$ variables left, and each matrix update equation involves one matrix vector product, $\mathbf{\Lambda} \mathbf{z}^n$, where the matrix is sparse and has at most two non-zero elements per row and is of dimension $2q \times 2q$. Further each update of a reduced model involves two matrix vector products, $\mathbf{F} \mathbf{u}^{n+1/2}$ and $\mathbf{E}^T \mathbf{z}^{n+1}$, where the matrices are dense and of dimension $2q \times 2$. One time step involves performing three of these matrix updates, whereas the regular FDTD simulation involves 4255 field variables with an update equation each. The amount of FLOP's needed was between 25, for $q = 10$, and 50, for $q = 5$, times smaller than with the standard FDTD method. By using a larger time step, the savings become proportionally larger.

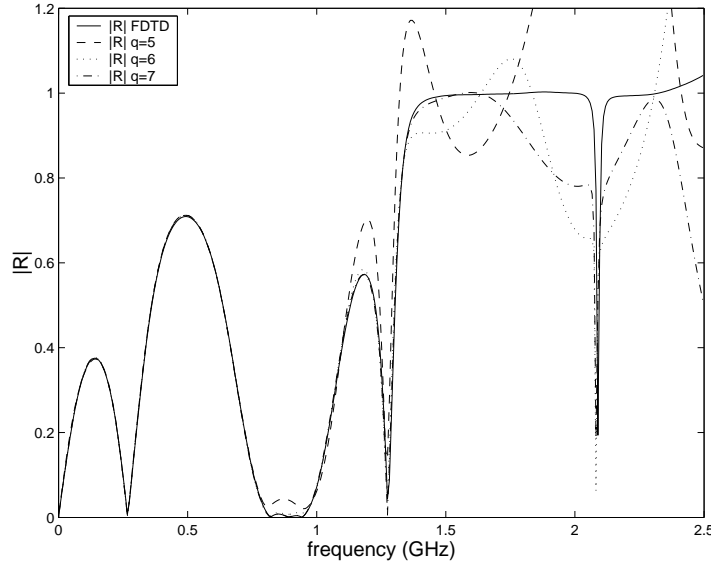


Figure 6.9: Reflection coefficient for $q = 5, 6, 7$ compared to the FDTD reference result, for a simulation where subdomain II is inserted twice.

This analysis was performed without considering the calculation of the reduced models. An implementation with the mathematical package Matlab, using the sparse routines available, showed that the reduction of all three systems required about one second, whereas timestepping was done in more than three seconds. Therefore ROM of moderate sized subdomains is not a limiting factor.

Since the subdomains are of a generic type: for each subdomain the left boundary field variables are magnetic fields and the right boundary field variables are electric fields, it is possible to reuse the reduced models. To illustrate this, the results of a simulation are shown where the middle subdomain, subdomain II, was inserted twice. Subdomain I and subdomain III remained unchanged. The leapfrog timestepping algorithm was as follows: at $t = n\Delta_t$ subdomain I and the second version of subdomain II is updated, at $t = (n + 1/2)\Delta_t$ the first version of subdomain II and subdomain III are updated.

In Fig. 6.9, the reflection coefficient for $q = 5, 6, 7$ is compared to the FDTD reference result. In Fig. 6.10, the transmission coefficient is compared. The same conclusions can be drawn for this case. This makes clear that the different subdomains and their ROM can be combined again leading to an even more efficient algorithm.

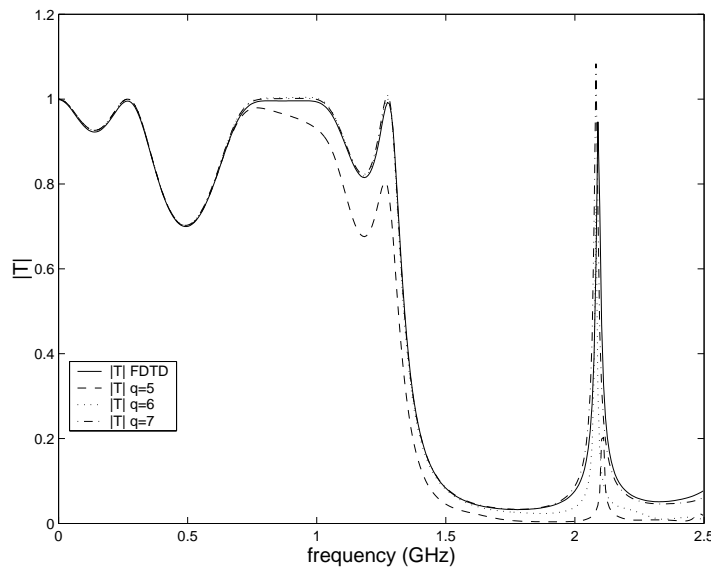


Figure 6.10: Transmission coefficient for $q = 5, 6, 7$ compared to the FDTD reference result, for a simulation where subdomain II is inserted twice.

3 SUBDOMAIN FDTD METHOD

3.1 Introduction

Over the years the finite-difference time-domain method (FDTD) has become a powerful numerical analysis tool in computational electromagnetics. A problem concerning the FDTD-method that has received a lot of attention in a number of ways, is the incorporation of electrically small obstacles in the simulation domain. Since small objects require a small space step for a correct discretization, often a lot smaller than what would be dictated by the smallest wavelength present, the memory requirements rise excessively if the standard FDTD-method is used. Corresponding with a smaller space step a smaller time step needs to be used as well, leading to excessive computing time.

One way to circumvent this problem is the use of subcell models. Based on the knowledge of the analytic local field behaviour around certain specific, widely used, small objects, one locally adapts the regular FDTD equations. Some twenty years ago this has already been done for a thin perfect electrically conducting (PEC) wire [5], where the in-cell inductance of the wire was the key factor linking the charge and current on the wire to the

fields. In [6] the $1/r$ field assumption of the tangential magnetic and the radial electric field near a thin wire were incorporated in the update equations. Later these subcell models were improved for wires with dielectric coatings [7], for wire end effects [8] and for bundles of wires [9]. A lot of researchers have also investigated the development of subcell models for thin slots. In [10] a thin-slot formalism was introduced that allowed modeling of arbitrarily narrow slots, based on the quasi-static behaviour incorporated in an in-cell slot capacitance. In [11], a thin-slot formalism was introduced that allowed modeling a slot with a depth of several cells. The update equations were derived using a Faraday's law contour integral approach. In [12], an integral-equation based thin-slot algorithm allowed to model slots with a very small depth by using an equivalent antenna. Subcell models were also developed for thin sheets: in [13] several methods for modeling thin dielectric sheets are compared. In [12] subcell models for thin conducting sheets are also compared, where the thickness of the sheet has to be smaller than the skin depth. A method for modeling good but not perfectly conducting sheets, for sheets thicker than the skin depth, is proposed in [14]. All these techniques tackle very specific and geometrically simple objects. More general approaches have also been proposed in [15] and for the acoustical FDTD method in [16]. There the FDTD update equations are adapted using correction factors which are obtained from the known static behaviour or from a quasi-stationary solution of the sub-wavelength region of interest. All the subcell models developed over the years originate from known or calculated static behaviour that is incorporated into the FDTD-algorithm, but as far as we know a general method has not yet been proposed.

The subdomain FDTD method, can be used to generate, in an automatic fashion, subcell models: small subgridded subdomains are replaced by a reduced model that captures the specific behaviour of the subdomain. This reduced model, is then in FDTD simulations used as a subcell model. In the examples hereafter, the term *subcell model* will sometimes be used to denote the reduced models. Most of the obtained results have been published, see [2], [17], [18].

3.2 Lossless free space.

All the examples that will be shown are TM: only the field components H_x , H_y and E_z exist, so the *O*-type field components are magnetic fields and the *P*-type field components are electric fields. First we start with a subcell model for a piece of empty space and we consider the spurious scattered field caused by this subcell model. As plane wave excitation we

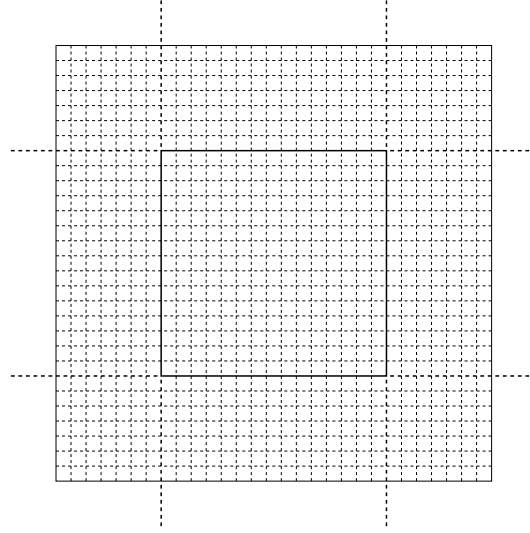


Figure 6.11: The subdomain used for the derivation of the free space sub-cell model, the refinement ratio is 15.

use the method established for a long time in [19], and which has proven to be very accurate when applying some modifications [20]. Here higher order cubic interpolation and the numerical phase velocity were used to determine the incident values of the plane wave. The sinusoidal signal was switched on using a Hanning window.

The subcell model is one coarse cell in size, the corresponding fine grid subdomain is shown in Fig. 6.11 and contains no special features. The bold lines correspond to the coarse grid and the other lines to the fine grid. The refinement ratio, Δ_c/Δ_f is $r = 15$. The fine grid boundary field variables use a first order accurate interpolation (2.36). The size of the different vectors is 2581 for \mathbf{x} , 8 for \mathbf{u} and 4 for \mathbf{y} . Based on this subdomain and using an order of approximation q , some subcell models were generated for $q = 1, 2, 4, 7$ and $f_{\max} = 100$ MHz. From this the dimension of the new vector \mathbf{z} (4.2), containing the internal variables of the reduced model, becomes $8q$. The resulting subcell model is then described by $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{C}}$ (dimensions $8q \times 8q$), $\tilde{\mathbf{B}}$ (dimensions $8q \times 8$) and by $\tilde{\mathbf{L}}$ (dimensions $8q \times 4$).

The spurious scattered field caused by the reduced order modeling and by the transition between the coarse and the fine grid has been calculated. The space step was $\Delta_c = 0.333$ m, the corresponding Courant time step is $7.85 \cdot 10^{-10}$ s. To assure stability, the time step had to be chosen smaller: $\Delta_t = 3.9 \cdot 10^{-10}$ s or about half the maximum Courant time step. The to-

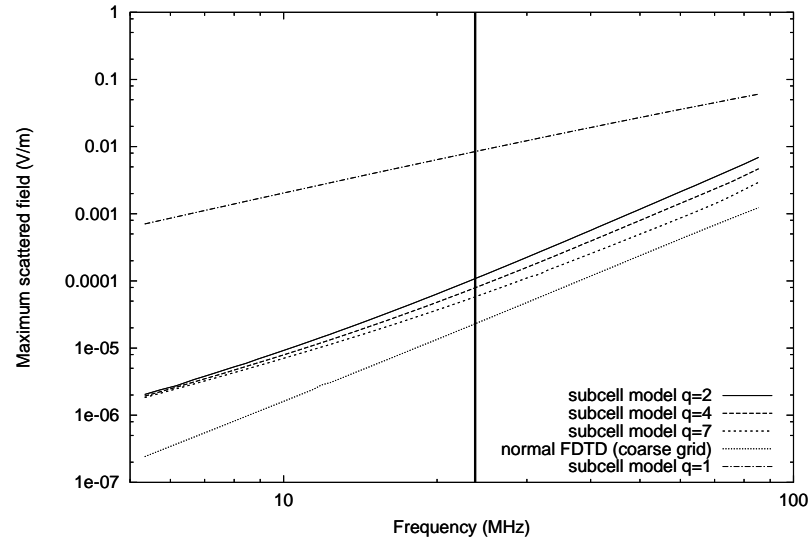


Figure 6.12: The maximum spurious scattered electric field amplitude as a function of frequency, for an angle of incidence of 36.9° .

tal field region was 3×3 coarse cells in size. Two plots were generated (Fig. 6.12-6.13). In each plot the maximum recorded scattered field was plotted, where the incident field has an amplitude of 1. In Fig. 6.12 the maximum scattered field amplitude has been plotted for each generated subcell model as a function of frequency. The angle of incidence (angle between x -axis and direction of propagation) was kept constant at 36.9° . The spurious scattered field for standard FDTD simulations was also added to illustrate the accuracy of the plane wave formalism. When $q \geq 2$, the spurious scattered field is small and the accuracy of the method can be observed. In Fig. 6.13 the maximum scattered field was plotted as a function of the angle of incidence, the frequency was kept constant at 23.7 MHz. Since the problem is symmetrical, only the interval between $0^\circ - 45^\circ$ was considered, the other angles can be derived from this. The case $q = 1$ was not included in the figure, due to poor accuracy. The angle of incidence does influence the accuracy, but for all angles, accuracy is good as long as $q \geq 2$. In both figures the bold vertical line shows the angle or the frequency that was kept constant in the other figure.

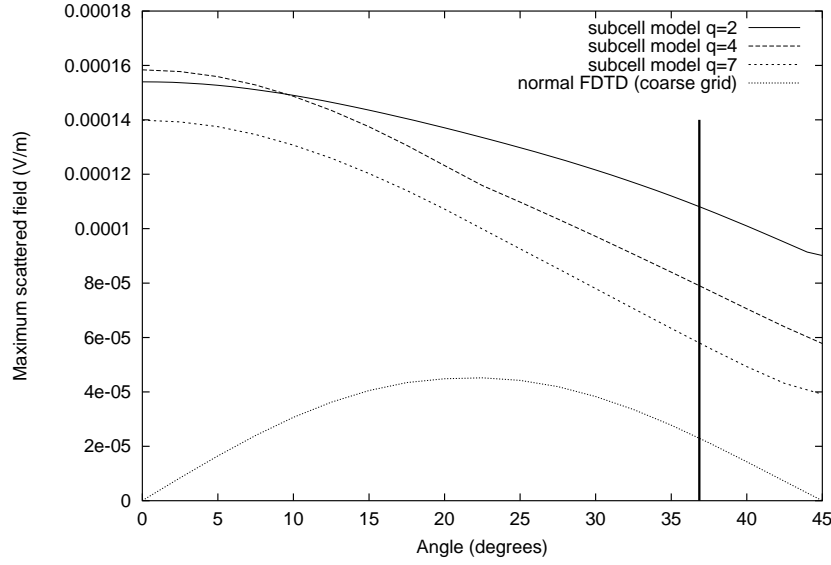


Figure 6.13: The maximum spurious scattered electric field amplitude as a function of the angle of incidence at $f = 23.7$ MHz.

3.3 Perfectly conducting thin wire.

In a second example a perfectly conducting wire was studied. A reduced model was calculated, which could then be compared to a subcell model previously derived in the literature [6], together with normal but finely meshed FDTD simulations. The radius of the wire we will consider is 0.4 mm, the space step Δ_c of the mesh is 1.7 mm, so the wire can easily be accommodated in one cell. The fine grid subdomain, containing the staircase approximation of the circular wire, that was used to generate the subcell model, is shown in Fig. 6.14. The wire was positioned in the middle of the cell. This is not necessary, but this was done to allow comparison with the method developed in [6]. The refinement ratio (Δ_c/Δ_f) is 17, so $\Delta_f = 0.1$ mm. Linear interpolation was used at the fine grid - coarse grid boundary (2.36). The original number of internal variables, i.e. the dimension of \mathbf{x} , was 3144 (known zero fields inside the wire are not part of \mathbf{x}). The dimension of the reduced order subcell model, equivalently the dimension of \mathbf{z} (4.2), is with $q = 1$, $pq = 8$, and the dimensions of the different matrices defining the subcell model are equal to those in the previous example. The factor α was chosen with $f_{\max} = 100$ MHz. Some very small artificial losses were added: $\sigma_e = 0.00005$ S/m and $\sigma_m = 0.00005$ Ω /m.

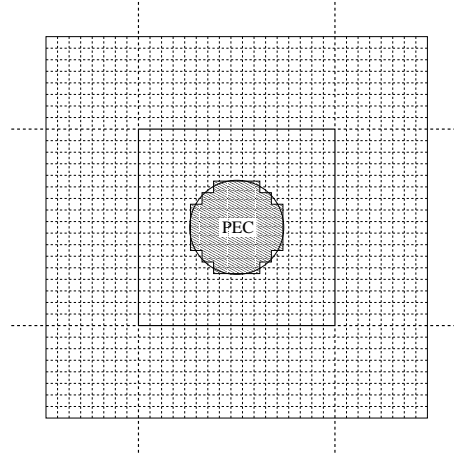


Figure 6.14: The subdomain used for the derivation of the PEC wire subcell model, the refinement ratio is 17.

Fig. 6.15 shows the configuration consisting of the wire, with at opposite sides the excitation and the electric field recording point. The line source was excited with a Gaussian pulse modulated at 1.25 GHz. This problem was simulated in 4 different ways:

- (i) the coarse grid ($\Delta = 1.7$ mm) and the subcell model introduced in [6],
- (ii) the subdomain FDTD method: the coarse grid ($\Delta = 1.7$ mm) and the subcell model based on a ROM of the subgridded subdomain,
- (iii) normal FDTD using a fine grid ($\Delta = 0.1$ mm) and staircase approximation of the PEC wire,
- (iv) normal FDTD using a very fine grid ($\Delta = 0.02$ mm) to get a better staircase approximation of the wire.

The time step was selected using the Courant limit in (i), (iii) and (iv). In simulation (ii), for reasons of stability, the time step was set at 0.55 times the Courant limit, or $\Delta_t = 2.2 \cdot 10^{-12}$, this is still more than eight times larger than the time step of the fine grid (iii). Even after a long time ($7.5 \cdot 10^6$ time steps) the tail of the results didn't show any sign of instability with this time step. In Fig. 6.16 the results are shown. It can be seen that the new subcell model shows better agreement than the old model [6] compared to the classic FDTD results. Fig. 6.17 shows the ratio of the frequency transformed electric field to the injected current for frequencies up to 10 GHz.

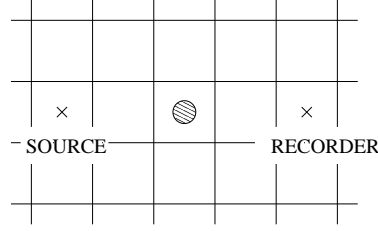


Figure 6.15: Test configuration for validation of generated PEC wire subcell models.

The same conclusions can be drawn: the generated subcell models show a better behaviour for the investigated wire radius, but the influence of the staircase approximation must not be underestimated. Another subcell model was also generated, with $q = 2$ (twice as much internal variables), but no difference was found between both subcell models, in this frequency band.

3.4 Dielectric thin wire.

The third example is similar to the previous one, but instead of studying the effects of a perfectly conducting wire, a dielectric wire is considered. In Fig. 6.18 the fine grid that was used for extracting the subcell model is shown. The different dimensions are: $\Delta_f = 0.1$ mm, $\Delta_c = 1.3$ mm and the wire radius is 0.9 mm. The relative dielectric constant of the wire is: $\epsilon_r = 10$. The size of the subcell model does not cover just one cell but covers an area of 2×2 cells. The dimensions of the original system are: $\dim(\mathbf{x}) = 4408$, $\dim(\mathbf{u}) = 12$ and $\dim(\mathbf{y}) = 8$. Linear interpolation was used to link the boundary fine grid field variables to the coarse grid field variables. The number of internal variables ($=\dim(\mathbf{x})$) is reduced to $12q$ internal variables ($=\dim(\mathbf{z})$), and $f_{\max} = 100$ MHz was used.

The generated subcell model was validated in the configuration of Fig. 6.19 for several values of q . The source was a line source, the field was recorded at the other side of a periodic dielectric structure consisting of nine dielectric wires. The source was excited by a modulated Gaussian pulse. Fig. 6.20 shows the results for three different subcell models (different orders of approximation q) simulated in a grid with $\Delta = 1.3$ mm. These results can be compared with the normal FDTD results, obtained using a fine grid ($\Delta = 0.1$ mm). The time step used in the simulations containing the subcell models was $1.4 \cdot 10^{-12}$ s or about 0.45 times the Courant

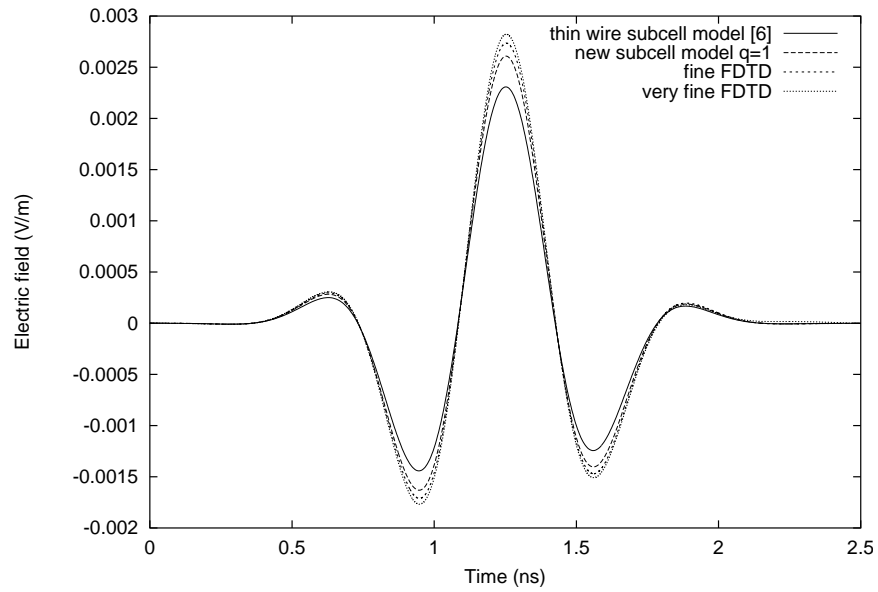


Figure 6.16: Recorded electric field: time domain results.

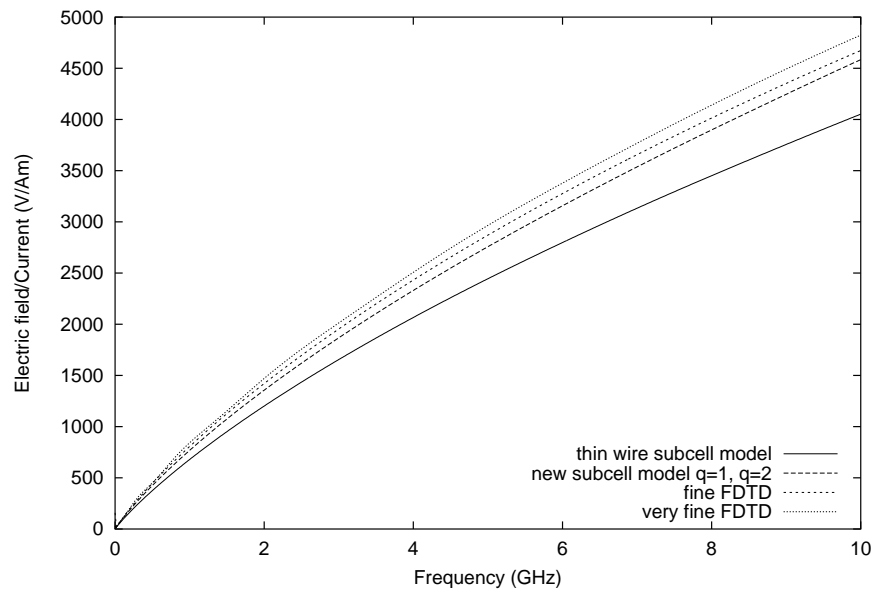


Figure 6.17: Amplitude ratio of electric field to line source current: frequency domain results.

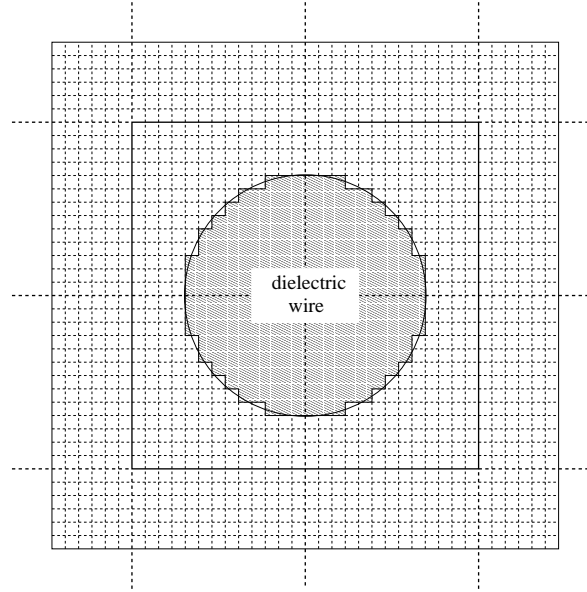


Figure 6.18: The subdomain used for the derivation of the dielectric wire subcell model, $r = 13$.

limit related to the coarse grid ($\Delta = 1.3$ mm). With this time step results showed no sign of instability even after $1.5 \cdot 10^7$ time steps. The curve with $q = 1$ agrees with the FDTD result only in a small frequency region up to ± 0.5 GHz. When a better model is used, $q = 2$, the results agree up to much higher frequencies: ± 7 GHz. For $q = 3$ both curves only start to deviate at about 11 GHz.

In Fig. 6.17 and Fig. 6.20 it can be seen that the frequency region where we get a good approximation extends far larger than f_{\max} . The parameter f_{\max} follows from theoretical grounds (see [4], [21]), but all numerical examples show that this parameter is too conservative.

3.5 L-shaped lossy dielectric object.

Up to now all examples were symmetric and had either no or only small losses, in this section a reduced model of a small non-symmetric lossy dielectric object is studied. The non-symmetric object is L-shaped and shown in Fig. 6.21. The parameters of the lossy dielectric are: $\epsilon_r = 10$ and $\sigma_e = 20$ S/m. Cell sizes are $\Delta_f = 0.1$ mm, $\Delta_c = 1.3$ mm and the subcell model is 2×2 cells in size. The skin depth associated with the loss is about

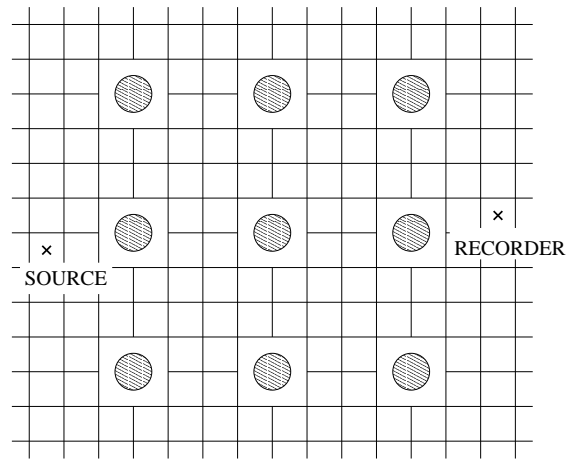


Figure 6.19: A 3×3 periodic structure test configuration for validation of dielectric wire subcell models.

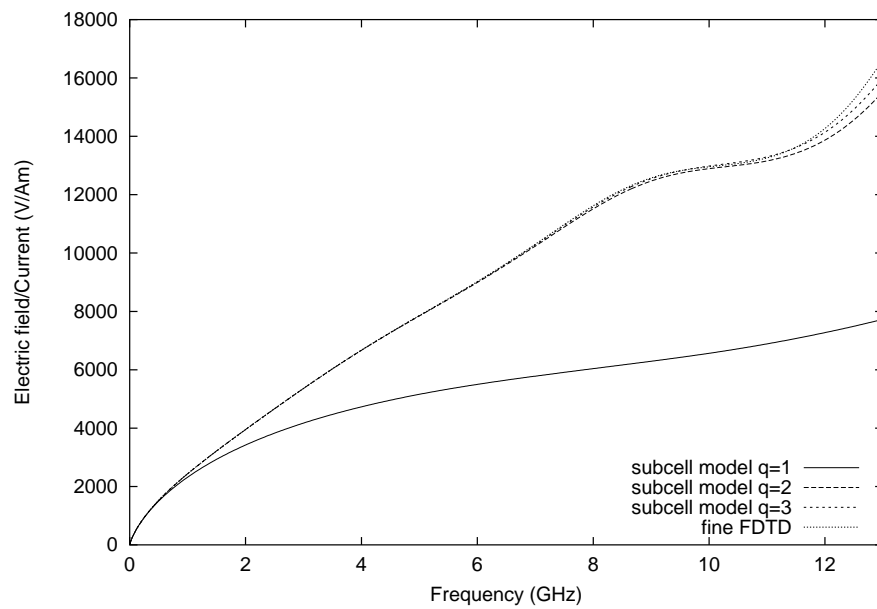


Figure 6.20: Amplitude ratio of electric field to line source current: frequency domain results.

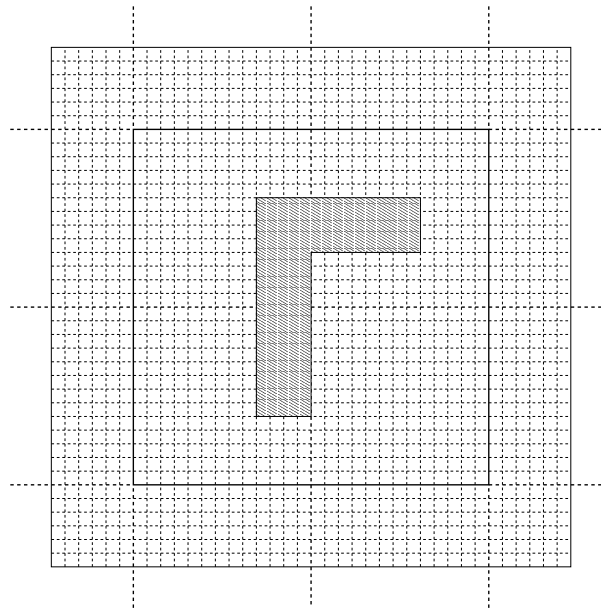


Figure 6.21: The subdomain used for the derivation of a subcell model of a lossy dielectric L-shaped object, $r = 13$.

$10\Delta_c$ at the maximum frequency of interest. The discretization, time step and dimensions of matrices and vectors are the same as for the previous example.

The configuration used for the simulation is shown in Fig. 6.22. In Fig. 6.23, the frequency results are shown, where the results based on a fine FDTD grid are used as a reference. It can be observed that with increasing order of approximation q , the frequency domain, where the model holds, becomes larger. For $q = 1$ this is up to 1.5 GHz, for $q = 2$ this is already 4.5 GHz and for $q = 3$ the model holds up to 6 GHz. We notice again that the subcell model holds from DC up to a certain frequency.

As an illustration of the power of the proposed method we compare in Table 6.2 the computational complexity (in flops) for this example as opposed to what was needed for the standard fine grid simulation. In Table 6.3 the memory savings are considered. The size of the grid as shown in Fig. 6.22 was used.

Table 6.2 illustrates that, if the same amount of simulated time is considered, even for the worst case $q = 3$, the computational savings are considerable: a factor of 180 to 460. The results in Table 6.2 are based on a time step equal to half the maximum time step of the coarse grid. When

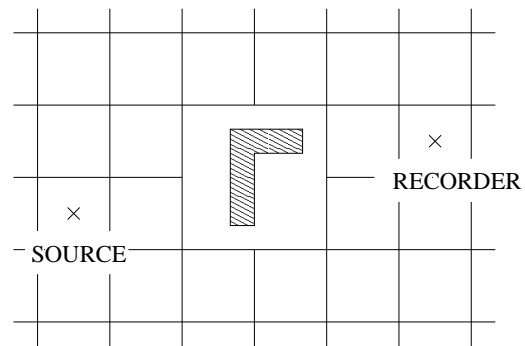


Figure 6.22: Test configuration for validation of lossy dielectric subcell model.

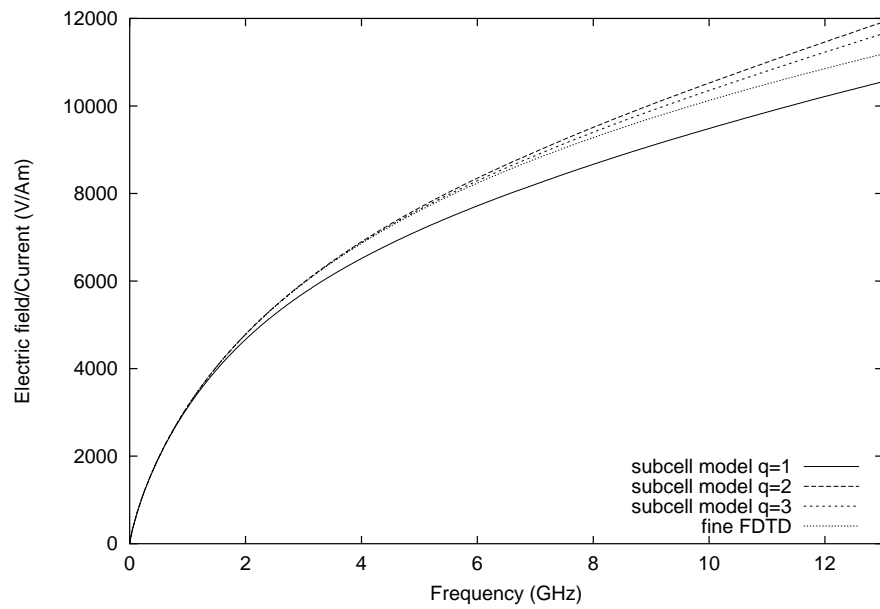


Figure 6.23: Amplitude ratio of electric field to line source current: frequency domain results.

memory is considered (see Table 6.3), a significant gain can also be observed: a factor of 30. Moreover, if the same model is used several times, as in the example with the thin dielectric wires, savings in memory are even larger. This is a consequence of the fact that for each repeatedly introduced subcell model, only a vector \mathbf{z} has to be added since the matrices in (6.3) remain the same and have to be stored only once.

Table 6.2: Comparison: computational complexity requirements.

| | subdomain FDTD | fine grid | |
|----------------------------|--------------------|------------------------|---------|
| cost of FDTD eq. | 48 + 48 + 80 | 8 268 + 8 216 + 16 224 | + and - |
| per time step | 24 + 24 + 20 | 4 134 + 4 108 + 4 152 | × and / |
| cost of eq. (6.3) | $12q \cdot 21 - 8$ | 0 | + and - |
| per time step | $12q \cdot 22$ | 0 | × and / |
| $\Delta_{t,c}$ / time step | ± 2 | 13 | |
| total / $\Delta_{t,c}$, | 924 | 425 204 | + and - |
| for $q = 3$ | 860 | 161 122 | × and / |

Table 6.3: Comparison: memory savings.

| | subdomain FDTD | fine grid |
|-------------------|----------------|-----------------------|
| FDTD equations | 48 + 48 + 40 | 8 268 + 8 216 + 8 208 |
| equation (6.3) | $12q \cdot 23$ | 0 |
| total for $q = 3$ | 828 | 24 692 |

3.6 Curved corner region.

The problem regions need not be surrounded by a single material, as was the case in the previous examples, materials can traverse the fine grid – coarse grid boundary, and consequently also the cut defining the subdomain. A subcell model of a curved corner region was considered. In Fig. 6.24, the subgridded subdomain is shown and Fig 6.25 shows the origin of the corner region: a material with three right corners and one curved corner. The material is a dielectric, $\epsilon_r = 10$, surrounded by vacuum. The refinement ratio is $r = 13$ and the fine grid region covers 3×3 cells. The cell sizes are $\Delta_f = 0.1$ mm and $\Delta_c = 1.3$ mm. This results in $\dim(\mathbf{x}) = 7905$,

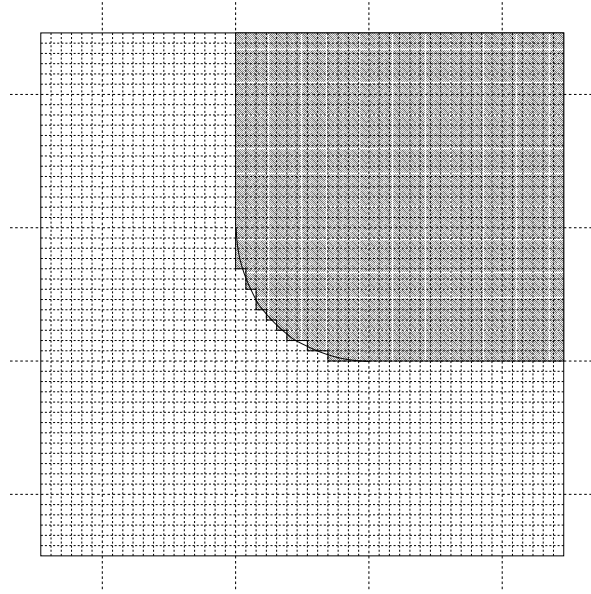


Figure 6.24: The fine grid subdomain used to calculate a subcell model of a curved corner, $r = 13$.

$\dim(\mathbf{u}) = 16$ and $\dim(\mathbf{y}) = 12$. The interpolation used by the fine grid boundary field variables was linear.

In Fig. 6.26 the frequency results are shown. The reference result was generated using the standard FDTD method and based on the fine grid, $\Delta = 0.1$ mm. The reduced model holds up to 1.5 GHz for $q = 1$, up to 4 GHz for $q = 2$ and up to 9 GHz for $q = 3$. The results confirm that the method is capable of incorporating materials crossing the fine grid - coarse grid boundary.

3.7 Influence of the interpolations.

In Chapter 2, it was shown that all spatial discretizations can be performed with second order accuracy. All the examples, up to now, used a linear interpolation. An interpolation for the fine grid boundary field variables, using three neighbouring coarse grid field variables results in second order accuracy (2.33). An example was elaborated based on the dielectric wire shown in Fig. 6.18. The test set up was similar to the configuration shown in Fig. 6.19, except that the distance between the wires has been increased by one coarse cell. This is shown in Fig. 6.27.

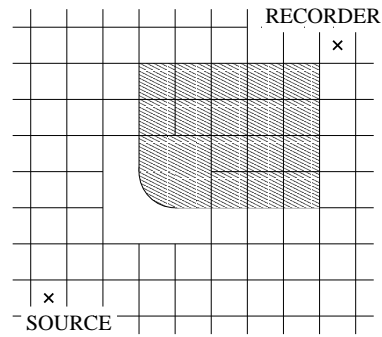


Figure 6.25: The configuration used to verify the curved corner subcell model.

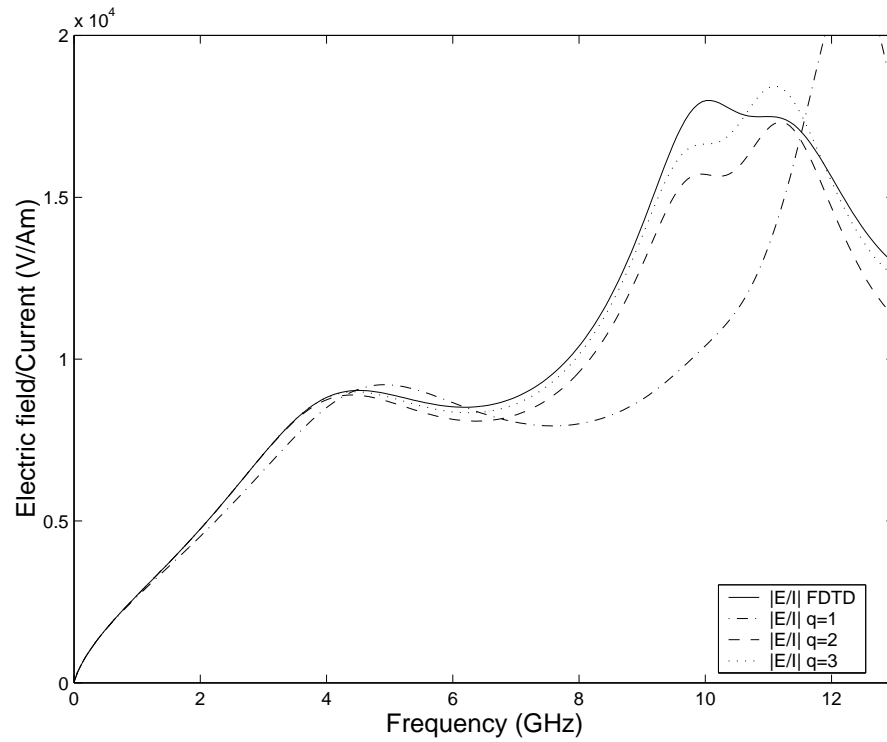


Figure 6.26: Amplitude ratio of the electric field to the line source current for the curved corner subcell model.

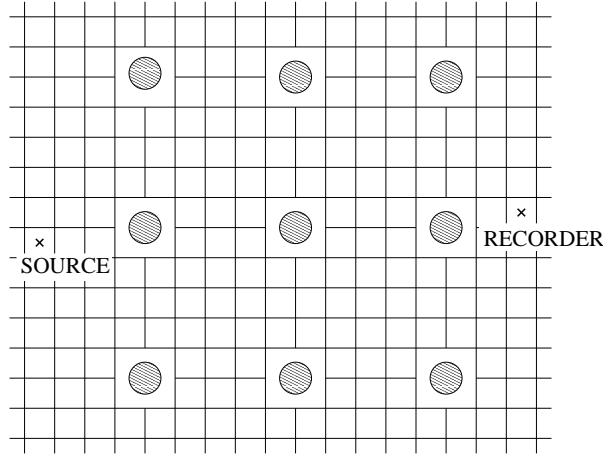


Figure 6.27: The configuration used to study the influence of the first order versus the second order accurate interpolations.

The parameters were identical as in Section 3.4, except that the time step could be chosen larger: $\Delta t = 18.4 \cdot 10^{-13} \text{s}$. For an order of approximation, $q = 3$, two subcells were calculated. For the first subcell model linear interpolation was used and for the second subcell model a second order accurate interpolation was used. Based on a fine grid FDTD simulation, the relative difference between both models was calculated and plotted on a logarithmic scale in Fig. 6.28. The figure shows that below $\pm 10 \text{ GHz}$ the results for both models are very accurate and in this frequency band it is not clear what model is best. Only around DC the relative difference becomes very high, due to the fact that the absolute values go to zero. For higher frequencies, however, when the results deviate from the standard FDTD result, the model based on the second order accurate interpolations is, albeit only slightly, more accurate.

3.8 Increasing time step

For all the examples considered up to now, the contour for terminating the subdomain and the fine grid coincided. It is however possible to expand the subdomain with some coarse grid cells. In Fig. 6.29, the dielectric wire model of Fig. 6.18 is shown again, this time with a small portion of the surrounding coarse grid. The examples, studied up to now, made no distinction between the contour used in the ROM step and the contour used in the time discretization step, each time cut C_1 was used. In this section, we

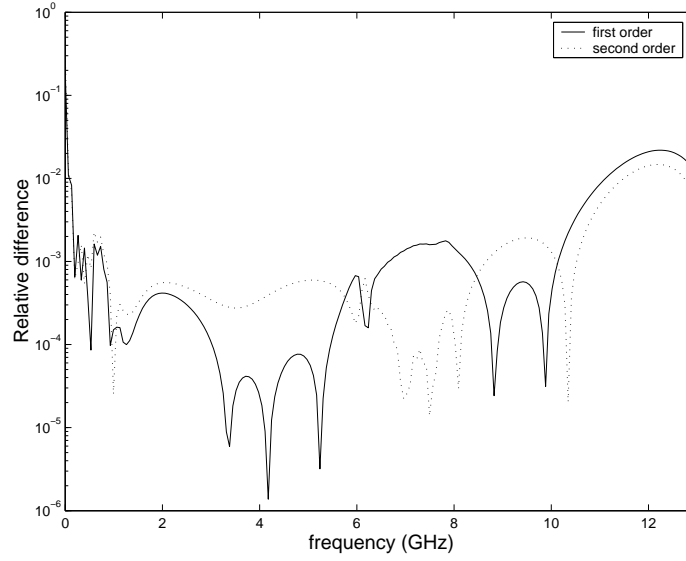


Figure 6.28: The relative difference between the reference result and reduced models, $q = 3$, where either first order accurate or second order accurate interpolations were used.

will compare results, by means of the configuration shown in Fig. 6.27, of a subcell model where in both cases contour C_1 was used in the ROM-step, but where, in the time discretization step, either contour C_1 , as used up to now, or contour C_2 was used. So, only the spatially reciprocal part, this is the part inside cut C_1 , was used in the ROM algorithm: the same ROM was used in both simulations but only the time discretization was performed differently. The subcell model based on the subdomain inside cut C_2 retains the field variables located between cut C_1 and C_2 without reduction. The subdomain is, after the ROM step of the algorithm, expanded from C_1 to C_2 . Both subcell models will be referred to based on the contour used in the time discretization step.

On the one hand, by expanding the subdomain in this way, the dimension of the subcell model increases. The maximum time step that can be used in the simulation, on the other hand, is higher, compensating for the increased subcell dimensions. In Table 6.4 the maximum time step, for which the simulation was stable after 5000 time steps, for the contour C_1 , was numerically examined for various values of q . This was done for the model based on the first and second order accurate interpolations. The dimensions of the matrices, which are present in the matrix update equa-

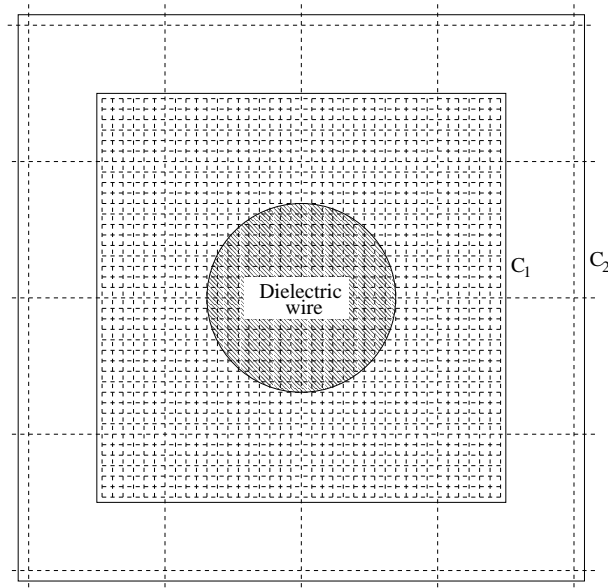


Figure 6.29: The dielectric wire model with two possible cuts for time discretization.

tion (6.3), are also shown. There are four input variables more than output variables.

In Table 6.5 the maximum time step, for contour C_2 , for which stable results were obtained is shown. In addition to the maximum time step the size of the matrices in (6.3) are indicated. The number of variables describing the model has increased by 40, accounting for the field variables between C_1 and C_2 . The input and output matrix have the same dimension in this case. For comparison it is noted that the coarse grid time step, or the Courant limit, is $3.066 \cdot 10^{-12}$ s.

Comparing both tables indicates first of all that the kind of interpolation used only plays a minor role as far as accuracy is concerned. What is more important is the observation that expanding the subdomain, from C_1 to C_2 , results in a higher maximum time step. The question whether to use C_1 or C_2 is problem dependent, since for small grids the larger time step cannot compensate for the extra computations needed by the subcell model. It also depends on how many times the same subcell model is used in one computation and how close the different subcell models are positioned from each other. Further it is observed that the maximum time step is always larger than half the Courant limit.

Table 6.4: Maximum time step for C_1 .

| q | $\Delta_{t,\max}$ 1 st order | $\Delta_{t,\max}$ 2 nd order | dim(Λ) | dim(F) | dim (E) |
|-----|-----------------------------------------|-----------------------------------------|------------------|----------------|---------------|
| 1 | 3.0×10^{-12} s | 3.0×10^{-12} s | 12×12 | 12×12 | 12×8 |
| 2 | 3.0×10^{-12} s | 3.0×10^{-12} s | 24×24 | 24×12 | 24×8 |
| 3 | 1.9×10^{-12} s | 1.9×10^{-12} s | 36×36 | 36×12 | 36×8 |
| 4 | 1.9×10^{-12} s | 1.8×10^{-12} s | 48×48 | 48×12 | 48×8 |
| 5 | 2.2×10^{-12} s | 2.0×10^{-12} s | 60×60 | 60×12 | 60×8 |
| 6 | 2.2×10^{-12} s | 2.0×10^{-12} s | 72×72 | 72×12 | 72×8 |
| 7 | 2.2×10^{-12} s | 2.0×10^{-12} s | 84×84 | 84×12 | 84×8 |
| 8 | 2.2×10^{-12} s | 2.0×10^{-12} s | 96×96 | 96×12 | 96×8 |

Table 6.5: Maximum time step for C_2 .

| q | $\Delta_{t,\max}$ 1 st order | $\Delta_{t,\max}$ 2 nd order | dim(Λ) | dim(F) | dim (E) |
|-----|-----------------------------------------|-----------------------------------------|------------------|-----------------|-----------------|
| 1 | 3.0×10^{-12} s | 3.0×10^{-12} s | 52×52 | 52×16 | 52×16 |
| 2 | 3.0×10^{-12} s | 3.0×10^{-12} s | 64×64 | 64×16 | 64×16 |
| 3 | 2.6×10^{-12} s | 2.6×10^{-12} s | 76×76 | 76×16 | 76×16 |
| 4 | 2.6×10^{-12} s | 2.6×10^{-12} s | 88×88 | 88×16 | 88×16 |
| 5 | 2.7×10^{-12} s | 2.7×10^{-12} s | 100×100 | 100×16 | 100×16 |
| 6 | 2.7×10^{-12} s | 2.7×10^{-12} s | 112×112 | 112×16 | 112×16 |
| 7 | 2.7×10^{-12} s | 2.7×10^{-12} s | 124×124 | 124×16 | 124×16 |
| 8 | 2.7×10^{-12} s | 2.7×10^{-12} s | 136×136 | 136×16 | 136×16 |

Another advantage of choosing C_2 is that the non-reciprocal part of the grid, the source of instability, is incorporated inside the subdomain, making it accessible to possible remedies for the loss of reciprocity. To illustrate the instability of the subcell models defined by C_2 , the eigenvalues of Λ were calculated and

$$\beta = \max(|\lambda_i|) - 1 \quad (6.4)$$

was used to indicate how much the largest eigenvalue of the subdomain was larger than one. This parameter can be seen as an indication for the late-time instability. The maximum time step, on the other hand, studied in Table 6.4 and Table 6.5 was an indication of a Courant like condition, see Fig. 5.12. In general β increases as q increases. It is therefore interesting to choose q as small as possible. Up to now no remedy for this possible

instability, that does not affect the accuracy too much, has been found. The examples illustrate that the method is useful as it is.

Table 6.6: Maximum eigenvalue for C_2 .

| q | β first order | β second order |
|-----|----------------------|----------------------|
| 1 | $6.7 \cdot 10^{-16}$ | $1.8 \cdot 10^{-15}$ |
| 2 | $1.8 \cdot 10^{-9}$ | $1.8 \cdot 10^{-9}$ |
| 3 | $5.4 \cdot 10^{-9}$ | $5.2 \cdot 10^{-9}$ |
| 4 | $6.6 \cdot 10^{-9}$ | $5.8 \cdot 10^{-9}$ |
| 5 | $2.1 \cdot 10^{-7}$ | $1.5 \cdot 10^{-5}$ |
| 6 | $5.6 \cdot 10^{-6}$ | $4.5 \cdot 10^{-6}$ |
| 7 | $2.1 \cdot 10^{-5}$ | $1.4 \cdot 10^{-5}$ |
| 8 | $2.2 \cdot 10^{-5}$ | $6.7 \cdot 10^{-6}$ |

For C_1 , β was $\sim 10^{-16}$, the relative floating point accuracy of the mathematical program and this regardless of q . This corresponds to the results of Chapter 5, where it was shown that, theoretically, each eigenvalue of Λ belonging to C_1 is one in absolute value, since the problem is lossless.

3.9 Photonic crystal waveguide

3.9.1 Configuration

Recently, photonic crystal structures have been studied abundantly and one of the most important tools for analysis is the FDTD-method. The sub-domain FDTD method can be used to model these kind of structures more efficiently. As a reference a paper [22], where sharp bends in a photonic crystal waveguide were investigated, is used. A periodic dielectric structure consisting of dielectric rods in air, where the dielectric rods are placed on a square lattice with lattice constant a is investigated (see Fig. 6.30). The dielectric constant of the dielectric material is $\epsilon_r = 11.56$ and the radius of the dielectric rods is $0.18a$. This implies that the crystal has a photonic band gap from $f = 0.302 \times c/a$ to $f = 0.443 \times c/a$. By removing a line of rods, a single guided mode can be created that can be used to guide light. In [22] it is shown, using the FDTD-method, that this crystal can be used to guide light around sharp corners. A critical point in using the standard FDTD method as an analysis tool is that a fine space step is required to accurately model the rods. This fine space step is a lot smaller than $\lambda_{\min}/10 - \lambda_{\min}/20$, usually considered adequate for FDTD-simulations, and results

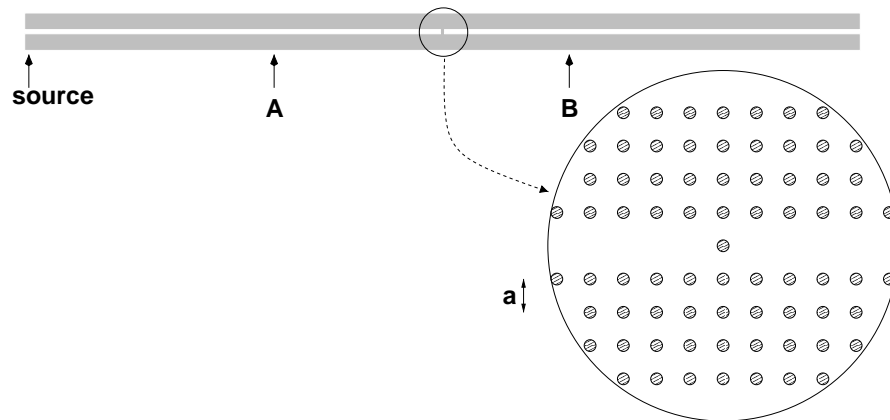


Figure 6.30: Photonic crystal waveguide with obstacle.

in high CPU-costs. The subdomain FDTD method avoids this by generating a subcell model of a dielectric rod in advance. It must be noted that the smaller the dielectric rods are, the more efficient the subdomain FDTD method becomes, since the standard FDTD method then requires a smaller space step and time step, whereas for the subdomain FDTD method, this is not the case.

The structure that was simulated was a long waveguide disturbed by an obstacle in the middle of it (see Fig. 6.30). The obstacle, causing the reflection and transmission of an incident pulse was an identical dielectric rod. The structure was limited to 4 rows of rods above and below the line defect. These 4 rows proved sufficient to confine the light into the waveguide and allows to model the structure with a standard FDTD-simulation using a very fine space step. The length of the waveguide and the observation points A and B were chosen to be able to clearly identify the incident, reflected and transmitted pulses. These observation points were chosen far enough from the edge of the computational domain to avoid disturbances from the absorbing boundary conditions. This all resulted in the following choices: the length of the waveguide was $201a$, observation A was chosen at $40a$ from the obstacle, observation B at $30a$ at the other side of the obstacle. In Fig. 6.31, the field amplitude recorded at observation point A is shown. The incident and reflected pulse can clearly be distinguished. At the end the spurious reflection due to the imperfect ABC can be seen.

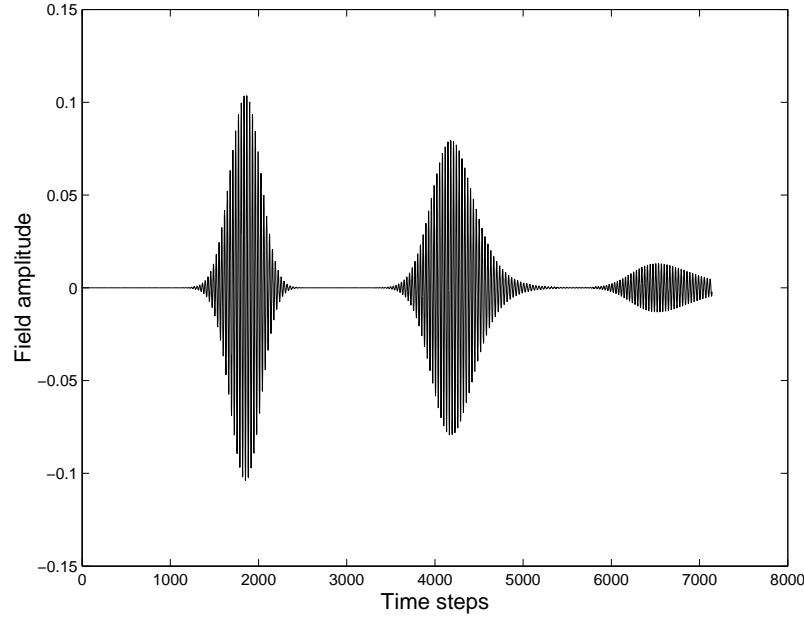


Figure 6.31: The time result of the field amplitude at observation point A.

3.9.2 Simulation

The waveguide problem as explained above was simulated in several ways. As reference simulation we use a standard FDTD simulation where $a = 52\Delta$.

The other simulations were performed with the proposed algorithm, where $a = 4\Delta$ was chosen. For this discretization the wavelength is between $13.2\Delta < \lambda < 9\Delta$ in the bandgap. The generalized subcell model was created from a finer grid with $r = 13$, i.e. $\Delta_{\text{fine}} = a/52$. In this way the discretization inside the subdomain is equal to the discretization used in the standard FDTD simulation. One fourth of the subdomain that was used to generate the model is shown in Fig. 6.32. Linear interpolation was used at the coarse grid - fine grid boundary, and this was also the boundary of the subdomain used in the time discretization step. The time step was such that $k = c_0\Delta_t/\Delta = 0.323$, where Δ is the coarse grid step. As can be seen in the results (Fig. 6.33), a higher order of approximation, q , meant a larger number of internal variables and computation cost, but also a broader frequency band where the model can be used. In Fig. 6.33 the results are shown for $q = 3, 4$. The result for $q = 3$ does not show good agreement, however, for $q = 4$ the result has improved considerably and shows very

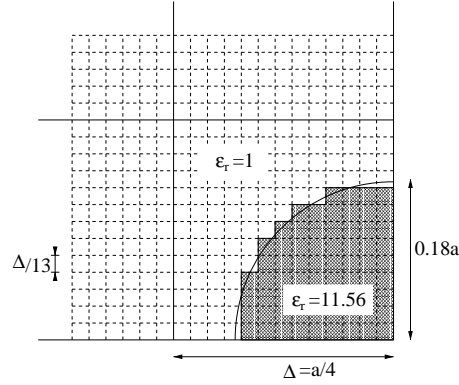


Figure 6.32: Upper left quadrant of the subgridded subdomain used to create a subcell model of a dielectric rod, $r = 13$.

good agreement for frequencies up to $f = 0.38a/c$. Unfortunately, for $q = 5$, the results were no longer useful due to late-time instability.

3.9.3 Efficiency

A comparison of efficiency can be made by considering one periodic lattice cell of dimension $a \times a$. In the standard FDTD simulation one lattice cell contained 52×52 Yee cells, with 3 field components in each Yee cell or 8112 field components. Since the space step was very small, the time step had to be chosen accordingly. In the subdomain FDTD simulations, one lattice cell of air, with size $a \times a$, located inside the line defect, contained 4×4 Yee cells or 48 field components. One lattice cell, of size $a \times a$, with a dielectric rod contained 32 regular field components and a subcell model of the dielectric wire, where the computational complexity is determined by the matrix-update equation (6.3). The dimension of \mathbf{z} was: 36 for $q = 3$ and 48 for $q = 4$. The time step that was used was about half the Courant limit. This means that the time step was six times larger than in the standard FDTD simulation.

Considering the computational complexity of simulating one lattice cell, containing a dielectric rod, for a fixed amount of time, the time step for the Courant limit of the coarse grid, it turns out that the FDTD method needs: $\pm 96\,000$ multiplications and $\pm 260\,000$ additions. The subdomain FDTD method, on the other hand, only requires ± 2200 multiplications and ± 2200 additions (for $q = 4$). This calculation shows the benefit for the worst case since, for the lattice cells where the line defect is located, no

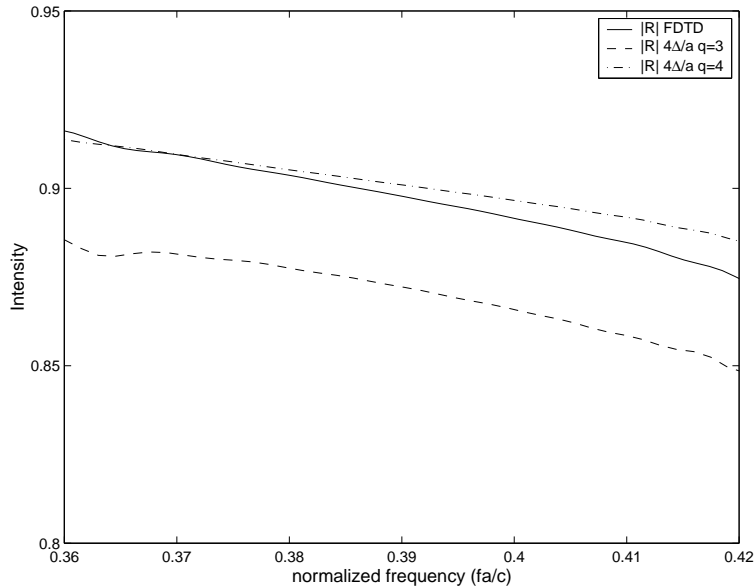


Figure 6.33: Amplitude of the reflection coefficient generated by standard FDTD, and by the subdomain FDTD method with $q = 3$ and $q = 4$.

subcell model needs to be used.

The savings in memory are even better because the matrices \mathbf{A} , \mathbf{E} and \mathbf{F} in (6.3) need to be stored only once, no matter how many times the model is used.

BIBLIOGRAPHY

- [1] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," *IEEE Microwave and Guided Wave Letters*, vol. 10, no. 8, pp. 301–303, Aug. 2000.
- [2] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Generation of FDTD subcell equations by means of reduced order modeling," *IEEE Trans. Antennas Propagat.*, accepted for publication.
- [3] B. Denecker, D. De Zutter, L. Knockaert, and F. Olyslager, "A higher level algorithm for 2D electromagnetic modelling using an FDTD grid," *IEEE Antennas Propagat. Symp.*, vol. 3, pp. 1340–1343, July 2000.

- [4] L. Knockaert and D. De Zutter, "Passive reduced order multiport modeling: The Padé-Laguerre, Krylov-Arnoldi-SVD connection," *AEU Int. J. Electron Commun.*, vol. 53, no. 5, pp. 254-260, 1999.
- [5] R. Holland and L. Simpson, "Finite-difference analysis of EMP coupling to thin struts and wires," *IEEE Trans. Electromagnetic Compatibility*, vol. 23, no. 2, pp. 88-97, May 1981.
- [6] K. R. Umashankar, A. Taflove, and B. Beker, "Calculation and experimental validation of induced currents on coupled wires in an arbitrary shaped cavity," *IEEE Trans. Antennas Propagat.*, vol. 35, no. 11, pp. 1248-1257, Nov. 1987.
- [7] J. J. Boonzaaij and C. W. I. Pistorius, "Radiation and scattering by thin wires with a dielectric coating — a finite-difference time-domain approach," *Microwave Opt. Techn. Letters*, vol. 5, no. 6, pp. 288-291, June 1992.
- [8] J. J. Boonzaaij and C. W. I. Pistorius, "Finite-difference time-domain field approximations for thin wires with a lossy coating," *IEE Proc.-Microw. Antennas Propagat.*, vol. 141, no. 2, pp. 107-113, Apr. 1994.
- [9] J. P. Bérenger, "A multiwire formalism for the FDTD method," *IEEE Trans. Electromagnetic Compatibility*, vol. 42, no. 3, pp. 257-264, Aug. 2000.
- [10] J. Gilbert and R. Holland, "Implementation of the thin-slot formalism in the finite-difference EMP code THREDII," *IEEE Trans. Nucl. Sci.*, vol. 28, pp. 4269-4274, Dec. 1981.
- [11] A. Taflove, K. R. Umashankar, B. Beker, F. Harfoush, and K. S. Yee, "Detailed FD-TD analysis of electromagnetic fields penetrating narrow slots and lapped joints in thick conducting screens," *IEEE Trans. Antennas Propagat.*, vol. 36, no. 2, pp. 247-257, Feb. 1988.
- [12] D. J. Riley and C. D. Turner, "Hybrid thin-slot algorithm for the analysis of narrow apertures in finite-difference time-domain calculations," *IEEE Trans. Antennas Propagat.*, vol. 38, no. 12, pp. 1943-1950, Dec. 1990.
- [13] J. G. Maloney and G. S. Smith, "A comparison of methods for modeling electrically thin dielectric and conducting sheets in the finite-difference time-domain (FDTD) method," *IEEE Trans. Antennas Propagat.*, vol. 41, no. 5, pp. 690-694, May 1993.

- [14] S. Van den Berghe, F. Olyslager, and D. De Zutter, "Accurate modeling of thin conducting layers in FDTD," *IEEE Microwave and Guided Wave Letters*, vol. 8, no. 2, pp. 75–77, Feb. 1998.
- [15] D. B. Shorthouse and C. J. Railton, "The incorporation of static field solutions into the finite difference time domain algorithm," *IEEE Trans. Microwave Theory Tech.*, vol. 40, no. 5, pp. 986–994, May 1992.
- [16] J. De Poorter and D. Botteldooren, "Acoustical finite-difference time-domain simulations of subwavelength geometries," *J. Acoust. Soc. Am.*, vol. 104, no. 3, pp. 1171–1177, Sep. 1998.
- [17] B. Denecker, F. Olyslager, D. De Zutter, and L. Knockaert, "2-D FDTD subgridding based on subdomain generation," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 288–290, May 2001.
- [18] B. Denecker, F. Olyslager, D. De Zutter, L. Klinkenbusch, and L. Knockaert, "Efficient analysis of photonic crystal structures using a novel FDTD-technique," *IEEE Antennas Propagat. Symp.*, vol. 4, pp. 344–347, June 2002.
- [19] A. Taflov, "Application of the finite-difference time-domain method to sinusoidal steady-state electromagnetic-penetration problems," *IEEE Trans. Electromagnetic Compatibility*, vol. 22, no. 3, pp. 191–202, Aug. 1980.
- [20] L. Gürel and U. Oğuz, "Signal-processing techniques to reduce the sinusoidal steady-state error in the FDTD method," *IEEE Trans. Antennas Propagat.*, vol. 48, no. 4, pp. 585–593, Apr. 2000.
- [21] L. Knockaert and D. De Zutter, "Laguerre-SVD reduced-order modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1469–1475, Sep. 2000.
- [22] A. Mekis, J. C. Chen, S. Fan, P. R. Villeneuve, and J. D. Joannopoulos, "High transmission through sharp bends in photonic crystal waveguides," *Physical Review Letters*, vol. 77, no. 18, pp. 3787–3790, Oct. 1996.

CHAPTER 7

Conclusions — Future Research

1 CONCLUSIONS

In this work two FDTD algorithms have been proposed. For both algorithms a clear distinction between spatial and temporal discretization has been made. The first algorithm has been called the subdomain FDTD method, and has the following characteristics:

- *Spatial discretization*: a problem space is spatially discretized, and for certain small specific areas subgridding is applied. The refinement ratio, i.e. the ratio of the space step in the coarse grid and the space step in the fine grid, is an odd integer. Refinement ratios up to 17 were used in the examples. All the discretizations are second order accurate, although, using simpler first order accurate discretization in the fine grid – coarse grid transition region, the quality of the results changed only slightly.
- *ROM*: to guarantee an efficient time discretization, a reduced order model (ROM) of the subgridded subdomain is generated. One important parameter in this step is the order of approximation. A higher order of approximation results in a larger ROM, i.e. more remaining internal variables, that is valid over a larger frequency domain.
- *Temporal discretization*: the reduced subdomain is time discretized using an implicit technique. For the remaining grid the explicit time discretization of the standard FDTD technique is used. Both time discretizations are second order accurate. A real diagonalization of the

resulting iteration matrix of the subdomain results in a further improved efficiency. In this way, the characteristic leapfrog time stepping, inherent to the standard FDTD algorithm, is maintained in a more general way: in an alternating fashion, electric fields and E -type subdomains at whole time steps, and magnetic fields and H -type subdomains at half time steps, are updated.

The subdomain FDTD method can be considered as a technique where, in an automatic and general fashion, subcell models are generated. The technique has a number of advantages: the generality of the approach, since it is based on subgridding, and the simplicity of the resulting algorithm once the reduced model has been obtained, resulting from a combination of implicit and explicit methods. Another advantage is that smaller features do not result in longer simulation times. For the standard FDTD method, a smaller feature requires a smaller space step and smaller time step, resulting in an increase to the third power in the 2D case. A final important aspect of the method is that it can be combined with most other FDTD techniques, especially with existing absorbing boundary conditions.

The second algorithm was called the generalized FDTD method and has the following characteristics:

- *Spatial discretization*: a problem space is spatially discretized using a uniform orthogonal grid. Since the grid is uniform and orthogonal, the discretization is second order accurate. The problem space is divided into a number of subdomains.
- *ROM*: to guarantee an efficient time discretization, a ROM of each subdomain is generated.
- *Temporal discretization*: each subdomain is time discretized with an implicit technique. A real diagonalization of the resulting iteration matrix of each subdomain results in a further improved efficiency. Finally, a leapfrog time stepping algorithm, comparable to a chess board, is used: the white and black subdomains are updated in an alternating fashion.

This approach is interesting for problems where the subdomains can be separated naturally, or in other words where subdomain boundaries can be considered to be ports, since in this way the number of variables at the subdomain boundaries can be limited.

For the subdomains in each of these algorithms a ROM needs to be generated. This ROM is then time discretized which involves the calculation of an inverse of a small matrix and a real block diagonalization. As ROM

technique we used the Laguerre-SVD technique [1] which was developed at the INTEC department. It has the following advantages in comparison to other ROM techniques: it is based on an expansion into orthogonal Laguerre functions and uses the robust singular value decomposition (SVD) to calculate a basis of the Krylov matrix.

The actual implementation of the algebraic steps behind the ROM, matrix inversions and block diagonalization were done using Matlab. The high level language and the readily available sparse routines result in a minimal implementation effort and fast computation of the reduced subdomain models. The computation time of the ROM was only a small fraction of the simulation time.

Although the mathematics involved are not always easy, the algorithms can be used efficiently with only investing limited amount of energy in the implementation. We implemented the reduced models in the FDTD simulator, "SimulateWorld", largely developed in a previous Ph. D. thesis [2].

In Chapter 5 a thorough analysis of the stability problem was performed. The analysis was not only restricted to the newly presented algorithms, but also considered the standard FDTD method and the ADI-FDTD method. In each of the characteristics of the new FDTD algorithms a cause of instability can hide:

- *Spatial discretization*: when spatial reciprocity is lost, it is no longer guaranteed that the system does not have poles in the right half plane.
- *ROM*: when the ROM algorithm is applied to a system, this system has to be written in a specific form where the system matrices are positive definite.
- *Temporal discretization*: for the subdomain FDTD method the conditional stability was investigated. It was shown that when the two previous steps did not create unstable poles, the stability is identical to the stability of a standard FDTD problem where the inside of the subdomain is removed. For the generalized subdomain FDTD method this result was extended and it was shown that only the fields at the boundary determine the stability.

A novel approach was developed to demonstrate the conditional stability of the standard FDTD method. This approach not only applies to a uniform lossless problem space, but also to a problem space containing any number of isotropic, either lossless or lossy, materials terminated by perfect conducting material. The unconditional stability of the ADI-FDTD method was also shown using this technique.

Finally, in Chapter 6, the new algorithms were validated with some numerical examples and the efficiency of the new approach was clearly demonstrated.

The work of this Ph. D. thesis was published in two international journal papers, [3, 4], in five articles in the proceedings of international conferences [5–10], in one abstract of an international conference [11] and in one article in the proceedings of a national conference [12]. Two publications, a first on stability and a second on photonic bandgap materials, are in preparation.

2 FUTURE RESEARCH

The algorithms developed and presented in this work have not yet reached a final point. A number of possible improvements, especially for the subdomain FDTD method, can be suggested:

- An improved ROM algorithm, which allows to derive a model near a specific frequency, is interesting for bandlimited problems. One such example is the photonic crystal structure studied in Chapter 6. Since this would result in a lower order of approximation to obtain similar accuracy, two consequences can be expected: first of all, since the model would be smaller, faster calculation and simulation times and secondly improved stability, since numerical examples showed that increasing the order of approximation results in reduced stability. Once such a ROM algorithm is available it can readily be implemented in our FDTD approaches.
- Improving the spatial discretization. A spatial discretization based on a non orthogonal grid or by using finite element techniques in the subdomain [13] can result in a spatially reciprocal subdomain model that is at the same time accurate. If this can be achieved, an accurate and guaranteed conditionally stable algorithm can be constructed.
- Extend the approach to 3D. Two possibilities exist: first of all the use of 2D models in 3D simulations. This is the approach of most current subcellular techniques. Secondly the generation of full 3D models. A major concern there is the reduction of the 3D models since the size for moderate sized subdomains is already very large. At this moment, promising results are being obtained by a new Ph. D. student, Gunther Lippens, where the reduction of 3D models of more than several hundreds of thousands field variables seems viable.

- The combination of the ADI-FDTD, another implicit method, to time discretize the subgridded subdomain and the standard FDTD method, in the surrounding grid, seems a promising new hybrid time discretization method. It needs to be investigated whether the approach would be as efficient, but the use of linear algebra and the ROM algorithms can be avoided in this way.

BIBLIOGRAPHY

- [1] L. Knockaert and D. De Zutter, "Laguerre-SVD reduced-order modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1469–1475, Sep. 2000.
- [2] S. Van den Berghe, *Object-oriented Electromagnetic Simulations with the Finite-Difference Time-Domain Method*, Doctoraatsthesis, Vakgroep Informatietechnologie, Faculteit Toegepaste Wetenschappen, Universiteit Gent, Gent, België, 1999.
- [3] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," *IEEE Microwave and Guided Wave Letters*, vol. 10, no. 8, pp. 301–303, Aug. 2000.
- [4] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Generation of FDTD subcell equations by means of reduced order modeling," *IEEE Trans. Antennas Propagat.*, accepted for publication.
- [5] B. Denecker, D. De Zutter, L. Knockaert, and F. Olyslager, "A higher level algorithm for 2D electromagnetic modelling using an FDTD grid," *IEEE Antennas Propagat. Symp.*, vol. 3, pp. 1340–1343, July 2000.
- [6] B. Denecker, F. Olyslager, D. De Zutter, and L. Knockaert, "2-D FDTD subgridding based on subdomain generation," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 288–290, May 2001.
- [7] B. Denecker, F. Olyslager, D. De Zutter, L. Klinkenbusch, and L. Knockaert, "Efficient analysis of photonic crystal structures using a novel FDTD-technique," *IEEE Antennas Propagat. Symp.*, vol. 4, pp. 344–347, June 2002.
- [8] L. Knockaert and B. Denecker, "Explicit reciprocity and reduced order modeling," *2001 URSI Int. Symp. on Electromagnetic Theory*, pp. 497–499, May 2001.

-
- [9] L. Knockaert, B. Denecker, and D. De Zutter, "Explicitly reciprocal reduced order modeling: Laguerre-SVD versus balanced realizations," *IEEE Antennas Propagat. Symp.*, vol. 2, pp. 556–558, June 2002.
 - [10] L. Knockaert, B. Denecker, D. De Zutter, and F. Olyslager, "Reduced order multiport modelling via Laguerre-SVD and its application to FDTD," *XXVIIth URSI General Assembly, Maastricht, The Netherlands*, pp. CD-ROM, Aug. 2002.
 - [11] B. Denecker, F. Olyslager, L. Knockaert, and D. De Zutter, "Efficient FDTD analysis of very large finite photonic crystal structures," *LEOS Benelux, Photonic Crystal Workshop, Ghent*, May 2002.
 - [12] B. Denecker, "The finite-difference time-domain method and reduced order modelling: Electromagnetics," *1e Doctoraatssymposium FTW, Gent*, p. 9, Dec. 2000.
 - [13] Y. Zhu and A. C. Cangellaris, "Macro-elements for efficient FEM simulation of small geometric features in waveguide components," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 12, pp. 2254–2260, Dec. 2000.

